Algoritmos y Estructuras de Datos

Mario Daniel Albarracín
Ingeniero en Electrónica
Universidad Tecnológica Nacional
Profesor Titular de Estructuras de Datos I
Universidad Argentina John F. Kennedy

Algoritmos y Estructuras de Datos

Mario Daniel Albarracín

Editor: Mario Daniel Albarracín

Diseño de tapa: Claudio Albarracín

Impreso en Argentina

A Patricia, Florencia y Nicolás
con todo mi amor.

Contenido

Capítulo 3. Estructuras de repetición

Capítulo 4. Ejercicios de propósito general

Capítulo 5. Vectores

Capítulo 6. Subprogramas

Capítulo 7. Matrices

Capítulo 8. Ejercicios combinados

Prólogo

El objetivo principal de este libro es introducir al estudiante en la teoría general de los principios de programación a partir de la construcción, desarrollo y aplicación de algoritmos y estructuras de datos internas.

Se intentó reflejar los avances acumulados en las últimas décadas, en los modelos y técnicas utilizados para desarrollar algoritmos; en nuestro caso tomando como herramienta de programación un pseudocódigo de características introductoras, fundamentalmente desde una perspectiva didactica.

De la misma manera que en trabajos realizados anteriormente, nuevamente se ha puesto especial preocupación en el aspecto pedagógico en cuanto al desarrollo y encadenamiento de los distintos temas tratados y en especial en el aumento del grado de complejidad de los ejercicios, producto de la experiencia acumulado en tantos años de dictado de la materia.

Sólo el proceso de construcción del conocimiento por parte del estudiante, permitirá incorporar los conceptos presentados, a partir de los temas tratados, del análisis de los ejercicios resuelto y del desarrollo de los ejercicios propuestos al final de cada capítulo.

A modo de ayuda para facilitar las actividades de estudio, al finalizar cada unidad se incorporó un resumen de los conceptos principales.

El texto está organizado en 6 capítulos y abarca hasta estructuras de datos internas.

Se decidió no incluir estructuras de datos externas (Archivos), por considerar que el grado de abstracción que presenta su estudio teórico, sólo es propicio acompañado con la práctica en la computadora y utilizando un lenguaje de programación, lo que no está dentro de los objetivos de este curso sino en una etapa posterior.

El primer capítulo introduce al lector en los conceptos básicos de la programación, por medio de la presentación de los principios y de las definiciones generales.

En los capítulos dos y tres analizan los distintos tipos de estructuras de decisión y repetición respectivamente, con ejemplos de diferentes formas de aplicación para cada caso.

El capítulo cuatro plantea distintos casos de problemas de propósito gene-

ral en donde se aplican las herramientas de programación aprendidas en capítulos anteriores.

En el quinto y en el sexto capítulo se estudia el primer tipo de estructura interna de datos: el vector, analizando algoritmos clásicos como los procesos de búsqueda de máximos y mínimos, o el ordenamiento de un conjunto de datos.

Dentro del capítulo siete se presentan las matrices, segundo tipo de estructura interna de datos, con ejercicios resueltos de los procesos más comunes de aplicación.

Para finalizar en el último capítulo, el ocho, se analizan una gran cantidad de ejercicios que combinan la utilización de vectores y matrices.

Agradezco la posibilidad recibida por parte de la Universidad Argentina John F. Kennedy, al permitirme ejercer la docencia en sus aulas y crecer día a día.

Al Licenciado Jorge Bonapace y al Licenciado Roberto Coscio por todo el apoyo brindado permanentemente a mi tarea. A Sarita Cuellar y Arnaldo Farías por su colaboración de siempre.

Por último, un agradecimiento especial a mis queridos alumnos.

El autor



Conceptos Básicos de Algoritmos

1.1 INTRODUCCIÓN

Podemos definir como algoritmo al conjunto finito y ordenado de pasos que nos permiten resolver un determinado problema.

Un algoritmo es un método de resolución, con el que podemos resolver sin ambigüedad el problema planteado, a partir de la secuencia de pasos elementales a aplicar.

Cuando hacemos una llamada telefónica, ponemos en marcha un conjunto ordenado y finito de pasos: tomar el teléfono, verificar el tono, etc.

Lo mismo ocurre cuando debemos reemplazar un neumático averiado. Ponemos en marcha un algoritmo colocar el freno de mano, tomar la rueda de auxilio, tomar las herramientas, etc.

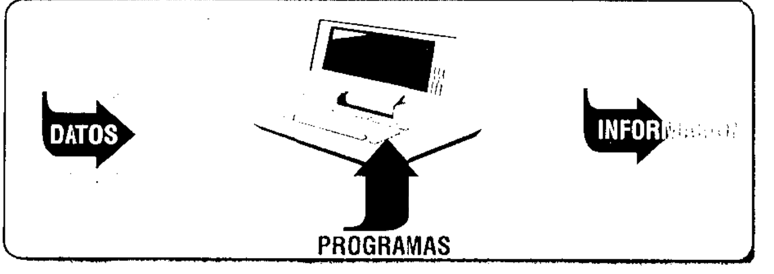
Pasamos gran parte de nuestro tiempo resolviendo problemas que se nos plantean a diario, es así que sin darnos cuenta vivimos desarrollando algoritmos.

1.2 RELACIÓN ENTRE ALGORITMO Y PROGRAMA

Si bien nuestro estudio está orientado a la resolución de problemas, específicamente nos centramos en el tipo de problema a resolver por una computadora. Se trata de proveer a la máquina del algoritmo necesario para que pueda llevar adelante los trabajos planteados. Se deben dar a la máquina las instrucciones que conforman el algoritmo.

Programar es escribir el algoritmo en un lenguaje común para el programador y la máquina.

Se denomina programa al conjunto finito y ordenado de instrucciones dadas a la computadora, para que esta procese los datos a partir de los cuales se obtendrá la información buscada.



Se puede observar en la figura 1, la diferencia conceptual entre dato e información.

Dato es el valor con el que contamos y que constituye una información no elaborada.

El resultado del proceso de los datos a partir de las instrucciones dadas, es la información elaborada.

1.3 LA COMPUTADORA COMO HERRAMIENTA DE TRABAJO

La computadora no es más que una herramienta de trabajo que nos permite tratar información en forma automática, constituyéndose de esta manera en una ayudante eficiente y veloz.

Podemos concluir en función de lo expresado, que el objetivo principal que tenemos que lograr es: que para un problema dado podamos diseñar una solución que pueda ser realizada por la computadora.

1.4 DISEÑO DE SOLUCIONES

Si bien diseñar una solución entra en el terreno de la creatividad y por lo tanto no existen dos soluciones iguales, podemos distinguir al menos dos etapas a cumplir:

- El análisis del problema.
- La programación propiamente dicha.

1.4.1 El análisis del problema

No se puede resolver un problema sin entender completamente el mismo.

¿Qué significa entender un problema?

Si bien no es fácil encontrar la respuesta en nuestra vida, no es tan complicado si acotáramos el problema en el entorno de una máquina. Es así, que diferenciamos al menos tres partes:

- Identificar los datos: saber con que contamos para la resolución.
- Determinar los resultados: tener claridad para interpretar que tenemos que resolver, que información queremos obtener.
- Explicitar las operaciones: encontrar la relación operacional entre los datos y los resultados a obtener.

En el siguiente ejemplo podemos identificar las tres partes:

Determinar el sueldo de un empleado, si el mismo varía de acuerdo a las horas trabajadas durante el mes.

- Datos: las horas trabajadas HT y el valor de la hora VH.
- Resultado: sueldo SDO.
- Relación operacional: SDO=HT x VH.

Cualquier problema puede tener diferentes formas de solución lo que implica que existen distintos tipos de algoritmos con ventajas y desventajas.

¿Cómo elegir el más adecuado?

Al menos distinguimos una serie de características a cumplir:

- Finito: el algoritmo debe finalizar, no puede tener un número de pasos ilimitado.

- Legible: debe ser fácil de leer y entender.

- Modificable: debe permitir su permanente actualización y modificación sin dificultad.

- Eficiente: la eficiencia debe comprender tres aspectos

- Rapidez: bajo tiempo de ejecución.

- Precisión: fiabilidad de los cálculos realizados.

- Aprovechamiento de la memoria al máximo.

- Modular: e problema debe divicirse en subproblemas o problemas más pequeños, de manera que la resolución de todos los subproblemas, determine la resolución del problema principal.

1.4.2 La Programación

Habíamos expresado anteriormente que programar es explicitar el algoritmo en la computadora.

¿Cómo damos las instrucciones a la máquina?

El problema está en que por ser la computadora una máquina limitada en velocidad, son pocas las operaciones básicas que puede llevar adelante:

- Operaciones aritméticas sencillas suma, resta, multiplicación y división.

- Operaciones lógicas sencillas: comparar dos valores y determinar si son iguales o cuál de ellos es el mayor o el menor.

- Almacenar y recuperar información.

La idea es poder escribir instrucciones a partir de estas operaciones sencillas de tal forma que dando un orden lógico a dichas instrucciones, generemos el programa que le permita a la computadora, resolver el trabajo que le

ue encomendado.

Durante el desarrollo del curso nos valdremos de un pseudocódigo o pseudolenguaje para escribir nuestros algoritmos.

Este pseudocódigo con finalidad didáctica, cumple en líneas generales con las características de lenguajes estructurados como: QBASIC,PASCAL, C, COBOL, etc.

Esto traerá aparejado una rápida adaptación de la teoría a la práctica sobre la computadora.

1.5 METODOLOGÍA DE LA PROGRAMACIÓN

En la resolución de nuestros algoritmos utilizaremos en forma combinada dos métodos: la programación modular y la programación estructurada.

- Programación modular: se basa en la realización de descomposiciones sucesivas del algoritmo inicial que describen el refinamiento progresivo del conjunto de instrucciones que conforman el programa. A este método se lo denomina descendente (TOP DOWN).

- Programación estructurada: el conjunto de instrucciones que constituyen el programa, se agrupan a su vez en subconjuntos denominados estructuras. Existen tres tipos de estructuras que serán abordadas oportunamente:

- estructura secuencial:	las instrucciones se ejecutan en el orden que se han codificado.
- estructura de decisión:	existen condiciones que de acuerdo a su cumplimiento o no, provocan la ejecución de fases diferentes del programa.
- estructura repetitiva:	la ejecución de un grupo de instrucciones se repite un número determinado de veces.

1.6 OBJETOS DE UN PROGRAMA

Se denominan objetos de un programa a todos aquellos manipulados por

las instrucciones. Se pueden distinguir en un objeto tres atributos: nombre, tipo y valor.

Se denomina identificador a las palabras creadas por el programador para dar nombre a los objetos.

1.7 TIPOS DE DATOS

Los datos pueden dividirse en tres grandes grupos:

•Datos numéricos: números con los cuales podemos realizar operaciones aritméticas. Pueden ser:

1- Enteros: todos los números positivos o negativos.

Ejemplo:

458, -217, -24.

2- Reales: todos los números positivos y negativos incluyendo a los que no son enteros.

Ejemplo:

-25.9 , 37.08 , 0.204

•Datos alfanuméricos que pueden ser de tres tipos:

1- Letras de la A, a la Z.

2- Numeros (no se pueden realizar operaciones aritméticas).

3- Caracteres especiales como guiones, paréntesis, etc.

• Datos booleanos: sólo pueden tener dos valores, verdadero y falso.

- no pueden leerse como datos.

- se forman a partir de los operadores especiales que analizaremos más adelante.

1.8 CONSTANTES

Son objetos cuyo valor permanece invariable a lo largo de la ejecución de un programa.

Las constantes pueden ser números enteros, números reales o caracteres alfanuméricos.

Ejemplo:

50	***constante entera.***
3.14159	***constante real.***
"Hola"	***constante alfanumérica.***

1.9 VARIABLES

Se denominan variables, aquellos objetos cuyo valor puede ser modificado a lo largo de la ejecución del programa.

En otras palabras, una variable no es otra cosa que una zona de la memoria de la computadora referenciada por un nombre, donde se puede almacenar el valor de un dato, que puede cambiar cuando lo deseemos.

El programador elige el nombre de la variable, el cual es aconsejable que no sea muy extenso y que tenga que ver con la información que contiene.

Una variable no es un dato, sino un área de la memoria que contiene al dato.

Para que una variable esté perfectamente definida, debe especificarse:

- Nombre.
- Tipo de dato que almacena.
- Valor inicial.

Como norma escribiremos nombres de variables de no más de ocho caracteres y en letra minúscula.

Como restricción, los nombres, obligatoriamente deben comenzar con una letra y no pueden tener espacios en blanco. El resto de los caracteres pueden ser números y agunos caracteres especiales.

Ejemplo:

sueldo
promedio
sueldo1
sueldo2
vol_max
c

Es importante remarcar que los lenguajes tienen palabras reservadas que no pueden ser utlizadas como nombres de variables.

Ejemplo:

IF
THEN
ELSE
WHILE
FOR
NEXT
PRINT

Según el tipo de datos que almacenan las variables pueden ser numéricas o alfanuméricas.

Como norma las diferenciaremos colocando el símbolo $ como último carácter de una variable alfanumérica.

Ejemplo:

a	***variable numérica***
a$	***variable alfanumérica***
nombre$	***variable alfanumérica***
sueldo	***variable numérica***

1.10 OPERADORES

Los operadores representan el tipo de operación a realizar entre los distintos objetos que conforman una instrucción.

Los operadores pueden ser aritméticos, alfanuméricos, relacionales y lógicos.

1.10.1 Operadores aritméticos

Permiten tratar números de manera aritmética.

La tabla a continuación, describe los distintos tipos de operadores, su significado y el order de prioridad.

Signo	***Operación***
+	***suma***
-	***resta***
*	***multiplicación***
/	***división***
^	***potenciación***

Ejemplo:

Expresión	***Resultado***
2^2	**4**
4*8	**32**
16/4	**4**
32+21	**53**
53-4	**49**

1.10.2 Operadores alfanuméricos

Se utilizan para unir datos alfanuméricos. En realidad es un solo tipo de operador que realiza la operación de concatenación.

Signo	Operación
+	concatenación

Ejemplo:

Expresión	Resultado
"Hola" + " "Pepe"	Hola Pepe
"3" + "." + "1416"	3.1416

Como se puede apreciar, concatenar equivale a unir los datos alfanuméricos.

1.10.3 Operadores relacionales

Estos operadores nos permiten comparar dos valores e indicar, si la relación planteada es verdadera o falsa.

Signo	Operación
>	mayor
<	menor
=	igual
>=	mayor o igual
<=	menor o igual
<>	distinto

Ejemplo:

Expresión	Resultado
30 > 10	verdadero
15 = 18	falso
17 <> 20	verdadero

1.10.4 Operadores lógicos

Combinan varios operadores relacionales con las reglas del álgebra de Boole, produciendo un valor final de la expresión, que será verdadero o falso.

Signo	Operación
OR	suma lógica
AND	producto lógico
NOT	negación

- OR: la expresión que forma es verdadera cuando alguna de las relaciones es verdadera.
- AND: la expresión que forma es verdadera cuando todas las relaciones planteadas son verdaderas.
- NOT: invierte el valor final.

Ejemplo:

Expresión	Resultado
9 > 4 OR 6 = 8	***verdadero***
10 < 20 AND 15 > 25	***falso***
NOT 10 > 3	***falso***

1.11 EXPRESIONES

Una expresión es la representación de un cálculo necesario para la obtención de un resultado.
Las expresiones pueden ser :

- Numéricas: producen resultados numéricos. Se construyen con operadores aritméticos.

- Alfanuméricas: producen resultados alfanuméricos. Se construyen con operadores alfanuméricos.

- Condicionales: producen resultados verdadero o falso. Se construyen con operadores relacionales y lógicos.

Cuando se realiza una expresión, la prioridad de evaluación de los operadores para determinar el orden en que se efectúan las operaciones, será conforme a la siguiente tabla:

Prioridad	Operación	Prioridad	Operación
1	***potenciación***	***5***	***relacionales***
2	***producto y división***	***6***	***negación***
3	***suma y resta***	***7***	***producto lógico***
4	***concatenación***	***8***	***suma lógica***

Es importante tener en cuenta las siguientes consideraciones:

- Si se desea cambiar el orden de prioridad, se deben utilizar paréntesis.

- En el caso de paréntesis anidados, se comienza siempre por los internos.

- Cuando dos operaciones tienen el mismo nivel de prioridad, la ejecución es de izquierda a derecha.

Tomemos por ejemplo la siguiente expresión:

14 * ((18 + 7) * 23 – 35) / 3 ^ 2 + 3 * 9

De acuerdo al nivel de prioridad se realiza de la siguiente forma:

1	***18 + 7 = 25***
2	***25 * 23 = 575***
3	***575 – 35 = 570***
4	***3 ^ 2 = 9***
5	***14 * 540 = 7560***
6	***7560 / 9 = 840***
7	***3 * 9 = 27***
8	***840 + 27 = 867***

1.12 ESTRUCTURA GENERAL DE UN PROGRAMA

Tal como se mencionó anteriormente, nos valdremos del pseudocódigo para escribir nuestros algoritmos. Utilizaremos un conjunto de palabras las cuales estarán sujetas a las reglas que implican que nuestras instrucciones deben ser interpretadas por la computadora.

Debemos tener presente que nuestros algoritmos van a ser desarrollados para poder programar la computadora, la que impone restricciones que debemos respetar.

El pseudocódigo nos permite en forma didáctica, hacer una primera aproximación a cualquier lenguaje real de programación, pues sigue las mismas reglas de escritura. Esto significa que se utilizará la metodología estructurada y modular, convirtiéndose en nuestra herramienta de diseño de algoritmos.

En general, un programa consiste en una secuencia de instrucciones que ha de procesar la computadora, con el objeto de obtener como resultado información a partir de un conjunto de datos de entrada.

Desde un punto de vista funcional, un programa se estructura en tres partes:

- Entrada de datos: está formada por todas las instrucciones que toman los datos objeto del programa desde un dispositivo externo (teclado, unidad de disco, etc), depositándolos en la memoria principal de la computadora, incluyendo la depuración o validación de los mismos.

- Proceso: es el conjunto de instrucciones que resuelven el problema a partir de los datos que han sido introducidos, dejando los resultados en la memoria principal. El dispositivo físico encargado de llevar a cabo esta tarea es la unidad central de proceso o CPU.

- Salida de resultados: la constituyen las instrucciones que hacen que los datos resultantes del proceso sean proporcionados al exterior por medio de algún dispositivo (monitor, impresora, etc).

De acuerdo a el tipo, podemos agrupar las instrucciones de la siguiente manera:

- Instrucciones de entrada: permiten ingresar los datos.

- Instrucciones de salida permiten mostrar los resultados.

- Instrucciones de asignación: son operaciones de asignación de valores a variables.

En resumen con nuestro pseudocódigo podemos manipular objetos en general: datos, variables, expresiones, etc.

1.13 INSTRUCCIONES DE ENTRADA

Toman datos del exterior almacenándolos en variables.

Sintaxis ***INGRESAR variable***

Ejemplo: ***INGRESAR numero***

En el ejemplo ingresamos desde el teclado un dato numérico que se almacenará en la variable de nombre " numero "

Si se desean leer varios datos, se pueden colocar en instrucciones consecutivas o bien en la misma instrucción separando las variables con coma.

Sintaxis: INGRESAR variable1,variable2,...,variableN

Ejemplo: ***INGRESAR alumno$, nota***

El ingreso de datos puede ser acompañado por comentarios aclaratorios que estarán delimitados por comillas.

Sintaxis: ***INGRESAR"comentario",variable***

Ejemplo: ***INGRESAR"Ingrese el nombre del alumno:",alumno$***

1.14 INSTRUCCIONES DE SALIDA

Presentan en pantalla o en impresora comentarios, constantes, contenidos de variables y resultados de expresiones.

- **Impresión de comentarios**

Sintaxis: ***IMPRIMIR "comentario"***

Ejemplo: ***IMPRIMIR"Buenas Noches"***

Los comentarios son constantes alfanuméricas.

- **Impresión de variables**

Sintaxis:	***IMPRIMIR variable***
Ejemplo:	***IMPRIMIR sueldo***

Se puede mostrar a partir de una única instrucción imprimir, el contenido de más de una variable.

Sintaxis:	***IMPRIMIR variable1,variable2,..., variableN***
Ejemplo:	***IMPRIMIR alumno$,nota***

- **Impresión de constantes**

Sintaxis:	***IMPRIMIR constante***
Ejemplo:	***IMPRIMIR 57***

- **Impresión de expresiones**

Sintaxis:	***IMPRIMIR expresión***
Ejemplo:	***IMPRIMIR suma / cantidad * 100***

Al igual que en el ingreso de datos, se pueden combinar en la salida comentarios con variables, etc.

Sintaxis:	***IMPRIMIR"comentario",variable***
Ejemplo:	***IMPRIMIR "El sueldo del empleado es:", sueldo***

1.15 INSTRUCCIONES DE ASIGNACIÓN:

Almacenan en una variable el valor de una constante, el resultado de una expresión de cálculo o bien el contenido de otra variable.

El cambio de valor de una variable se hace mediante la asignación, utilizando como operador el símbolo **=** que no expresa en este caso una igualdad: el valor a la derecha del símbolo se asigna a la variable de la izquierda.

Sintaxis:	***variable = constante***
Ejemplo 1:	***cantidad = 3***
Ejemplo 2:	***nombre$ = "María Florencia"***
Sintaxis:	***variable = expresión***
Ejemplo 3:	***promedio = (nota1 + nota2 + nota3) / 3***
Sintaxis:	***variable1 = variable2***
Ejemplo 4:	***a = b***

Se podría tener una instrucción de este tipo: **c = c + 1,** pues como se dijo la asignación no es una igualdad ; a la variable **c** se le asigna el valor que contiene incrementado en uno.

1.16 ESRUCTURA SECUENCIAL

La estructura secuencial está formada por un conjunto de instrucciones, que se ejecutan una a continuación de la otra, conformando una secuencia.

En realidad todo programa es una gran estructura secuencial donde se intercalan estructuras de decisión y repetitivas para controlar el flujo de los datos. En principio nos limitaremos a analizar estructuras secuenciales en su forma más simple: instrucciones de entrada, de asignación y de salida.

Los programas que realizaremos, tendrán la siguiente estructura:

- ENTRADA
- PROCESO
- SALIDA

Todos los ejercicios deben estar encabezados por una instrucción de arranque: COMIENZO; y finalizan con una orden de detención: FIN.

A continuación desarrollaremos un conjunto de ejercicios que nos permitirán comprender mejor este tipo de estructura.

Ejercicio 1

Dados dos números cualesquiera sumarlos.

```
COMIENZO
INGRESAR "Ingrese dos números:", a, b
suma = a + b
IMPRIMIR "La suma es", suma
FIN
```

En este primer ejercicio podemos ver las tres instrucciones mencionadas anteriormente. Se ingresan los datos con INGRESAR, se asignan a la variable suma y se muestra el resultado con la instrucción IMPRIMIR.

Como en todos los ejercicios que analizaremos podemos apreciar como está delimitado el programa por COMIENZO y FIN.

Ejercicio 2

Dados tres números, sumarlos de a dos.

```
COMIENZO
INGRESAR "Ingrese tres números:", a, b, c
suma1 = a + b
sumatot = suma1 + c
IMPRIMIR "La suma es:", sumatot
FIN
```


Observamos como dos asignaciones componen las instrucciones de proceso.

Ejercicio 3

Dados los tres lados de un triángulo, hallar el perímetro.

```
COMIENZO
INGRESAR "Ingrese el primer lado del triángulo:", a
INGRESAR "Ingrese el segundo lado del triángulo:", b
INGRESAR "Ingrese el tercer  lado del triángulo:", c
perimet = a + b + c
IMPRIMIR"El perímetro es:", perimet
FIN
```

A diferencia del ejercicio anterior, ahora se ingresan las tres variables con tres instrucciones INGRESAR. Cualquiera de las dos formas en principio es correcta; será la característica del ejercicio la que determinará cual variante usar.

Ejercicio 4

Dadas la base y la altura de un triángulo, hallar el área.

```
COMIENZO
INGRESAR "Ingrese el valor en cm. de la base:", b
INGRESAR "Ingrese el valor en cm. de la altura:", a
area = b * a / 2
IMPRIMIR "El área del triángulo es:", area, "cm"
FIN
```

Podemos apreciar en la impresión, como la variable numérica area, está ubicada entre dos cadenas de caracteres: "El área del triángulo" y "cm".

Ejercicio 5

Dados los lados de un rectángulo, calcular el perímetro, la superficie y la diagonal.

```
COMIENZO
INGRESAR "Ingrese el valor en cm. del lado mayor:", may
INGRESAR "Ingrese el valor en cm. del lado menor:", men
perim = (may + men) * 2
sup = may * men
diag = (may ^ 2 + men ^ 2) ^ (1 / 2)
IMPRIMIR "El perímetro es:", perim, "cm"
IMPRIMIR "La superficie es:", sup, "cm"
IMPRIMIR "La diagonal es:", diag, "cm"
FIN
```

Por primera vez se utiliza el símbolo ^, para representar la potenciación como operación aritmética. Por otra parte tendremos tres instrucciones IMPRIMIR para la salida de resultados que representan informaciones distintas.

Ejercicio 6

Dado un valor en kilómetros, expresarlo en metros.

```
COMIENZO
INGRESAR "Ingrese un valor en kilómetros:", km
mts = 1000 * km
IMPRIMIR "El valor en metros es:", mts
FIN
```

Nuevamente nos encontramos con una estructura secuencial muy simple donde podemos identificar rápidamente las instrucciones de entrada de datos, asignación de un cálculo y salida de resultados.

Ejercicio 7

Calcular el sueldo de un empleado, conociendo la cantidad de horas trabajadas. El valor de la hora es de 5 $.

```
COMIENZO
INGRESAR " Ingrese la cantidad de horas trabajadas:", horas
sueldo = horas * 5
IMPRIMIR "El sueldo es:", sueldo, "pesos"
FIN
```


Este ejercicio responde a la misma estructura que el anterior.

Ejercicio 8

Calcular el sueldo de un empleado, conociendo la cantidad de horas trabajadas y el valor que gana por hora.

```
COMIENZO
INGRESAR " Ingrese la cantidad de horas trabajadas:", horas
INGRESAR "Ingrese el valor de la hora:", valhora
sueldo = horas * valhora
IMPRIMIR "El sueldo es:", sueldo, "pesos"
FIN
```

Vemos una resolución muy similar a la del ejercicio 7, con la salvedad que ahora el valor de la hora es un dato a ingresar desde afuera del programa.

Ejercicio 9

Un comercio de venta de heladeras paga por mes a los vendedores, 500$, más una comisión del 10% del total de las ventas que realice. Si conocemos para el vendedor, la cantidad de heladeras vendidas y el importe de la heladera (se comercializa sólo un modelo).¿ Cuál será el sueldo del vendedor ?

```
COMIENZO
INGRESAR "Ingrese la cantidad de heladeras vendidas:", canthela
INGRESAR "Ingrese el importe de a heladera:", importe
sueldo = 500 + canthela * importe * 0.1
IMPRIMIR "El sueldo es:", sueldo, "pesos"
FIN
```

Por primera vez , calculamos un porcentaje, en este caso del 10%. Dicho valor esta expresado por el 0.1 que multiplica a las variables importe y canthela dentro del cálculo del sueldo.

RESUMEN

Algoritmo: secuencia ordenada de pasos que permite la resolución de un problema.

Programa: explicitación del algoritmo en una computadora.

Dato: es toda información sin elaborar que utiliza la computadora.

Información: es el resultado del proceso de los datos.

Etapas en el diseño de soluciones:

- análisis del problema
  Se pueden diferenciar tres partes:

  1-identificar datos.
  2-determinar resultados.
  3-escribir las operaciones que vinculen los datos con los resultados

Se pueden distinguir una serie de características a cumplir en la elaboración de un
problema : finito , legible , modificable , eficiente y modular.

- la programación propiamente dicha:
  Las operaciones básicas que puede llevar adelante una computadora son:

  1-operaciones aritméticas sencillas.
  2-operaciones lógicas sencillas.
  3-recuperar y almacenar información.

Pseudocódigo: descripción de un algoritmo utilizando una combinación de instrucciones escritas en lenguaje común simple de un lenguaje de programación.

Programación modular: descomposición del problema en subproblemas simplificando de esta forma la resolución

Programación estructurada: todo programa está organizado según tres tipos de estructura:

- secuencial: las instrucciones se ejecutan en el orden que se han codificado.

- decisión: de acuerdo al cumplimiento o no de relaciones se ejecutan grupos de instrucciones diferentes.

- repetición: la ejecución de un grupo de instrucciones se repite un número determinado de veces.

Objetos: todos aquellos manipulados por las instrucciones.

Atributos: un objeto posee como atributos el nombre , el tipo y el valor.

Tipos de datos: numéricos , alfanuméricos y booleanos.

Variables: un área de memoria de la computadora que contene un dato.

Tipo de variables: numéricas , alfanuméricas y booleanas.

Constante: valor que permanece invariable a lo largo del proceso.

Operadores: se utilizan para construir expresiones y pueden ser:

- aritméticos.
- alfanuméricos.
- relacionales.
- lógicos.
- paréntesis.



Ejercicios Propuestos



1. Indicar el valor de **x** para cada una de las siguientes expresiones:

a) x = (4 + 6) * 12
b) x = (10 + 22) / 4
c) x = (20 + 7) / 3 ^ 2
d) x = 5 * 5 + 10 * (8 - 6)

2. Para a = 5 , b = 3 y c = 2 , indicar el resultado final de las siguientes expresiones:

a) a - b + c
b) c * b + c ^ a
c) a * b / (b + c)
d) (a + b + c) * b

3. Indicar el valor final de **a** , **b** , **c** y **d**

a)
a = 3
b = 2
c = a + b
a = c + b
b = c

b)
a = 2
b = 6
c = b - a
d = a + b
a = c + 2 * d
b = 5 * d - c ^ 2
c = a * b
d = b + c

4. Indicar si son verdaderas o falsas las siguientes expresiones:

a) "25" + "25" = "50"
b) "25" + "25" = "2525"
c) "espacio" = "espacio "
d) "espacio" = "espacio"
e) 32 + 18 = 50
f) 53 < 24 AND 12 = (6 + 4 + 2)
g) 24 < 12 OR 9 = 3 ^ 2
h) NOT 24 < 12

5. Realizar un programa que permita hallar la longitud de la hipotenusa de un triángulo. Se tiene como dato las longitudes de los catetos.

6. El precio del pasaje para un vuelo es de 680$ en clase turista y se aplica un incremento de 30% en primera clase.
Se desea saber la recaudación obtenida en un vuelo.

capítulo 2

Estructuras de Decisión

2.1 INTRODUCCIÓN

Controlan el flujo de ejecución al seleccionar que grupo de instrucciones deben ejecutarse. De esta manera mejora el funcionamiento del programa al poder realizarse un mayor número de tareas.

Realizan acciones alternativas, pues la ejecución de una instrucción o grupo de instrucciones depende de si se cumple o no, una o varias condiciones.

2.2 INSTRUCCIONES DE DECISIÓN

Se utiliza la instrucción SI y la respuesta puede ser verdadera o falsa, es decir sí o no.

Ejemplo: Si llueve, tomaré un taxi.

La realización de la acción está supeditada a que se cumpla la condición. Existen distintos tipos de formato para instrucciones de decisión:

- Decisión simple
- Decisión doble
- Decisión en bloques
- Decisión múltiple

2.2.1 Instrucción de decisión simple:SI......ENTONCES......

Sintaxis:	***SI condición ENTONCES instrucción***
Ejemplo:	***SI x > 0 ENTONCES IMPRIMIR x***

La computadora primero examina la condición. Pueden suceder dos cosas:

- Si se cumple realiza la instrucción que está a la derecha de ENTONCES y continúa ejecutando las que están fuera del SI.
- Si no se cumple, no entra en el SI y ejecuta las instrucciones siguientes.

Se puede poner más de una condición, siempre y cuando estén unidas por los operadores lógicos OR, AND y NOT.

Sintaxis:

SI condición1 operador lógico condición2 ENTONCES instrucción

2.2.2 Instrucción de decisión doble: SI......ENTONCES.......SINO.......

Es muy común tener que realizar dos procesos completamente distintos, dependiendo de si se cumple o no la condición.

Ejemplo: Si llueve, tomaré un taxi; sino, iré caminando.

Trasladando este tipo de estructura a pseudocódigo, tenemos la instrucción S..........ENTONCES..........SINO

Sintaxis:	***SI condición ENTONCES instrucción1 SINO instrucción2***
Ejemplo:	***SI a >= b ENTONCES IMPRIMIR a SINO IMPRIMIR b***

Es decir:

- Si se cumple la condición realiza la instrucción de imprimir el contenido de la variable a.
- Si no se cumple la condición, imprime el contenido de la variable b.

Veamos a continuación algunos ejemplos:

Ejercicio 1

Dados dos números distintos, imprimir el mayor.

```
COMIENZO
INGRESAR "Ingrese dos números:", num1, num2
SI num1 > num2 ENTONCES IMPRIMIR num1 SINO IMPRIMIR num2
FIN
```

En el ejemplo se comparan las dos variables numéricas num1 y num2; la impresión de la mayor esta condicionada en la utilización de la instrucción S ENTONCES SINO.

Ejercicio 2

Dado un número decir si es cero.

```
COMIENZO
INGRESAR "Ingrese un número:", num
SI num > 0 ENTONCES IMPRIMIR "cero" SINO IMPRIMIR "no es cero"
FIN
```

Este es un ejemplo similar al anterior, con la diferencia que se condiciona ahora la impresión de un mensaje como resultado final.

2.2.3 Instrucción de decisión en bloques:

Se utiliza cuando las acciones siguientes a ENTONCES o SINO están compuestas por varias instrucciones.

Sintaxis:

```
SI condición ENTONCES

        instrucción(es)
SINO

        instrucción(es)

FIN SI
```

Ejemplo:

```
INGRESAR a,b,c
SI a < 0 ENTONCES
      suma = a + b + c
      IMPRIMIR "La suma es :",suma
SINO
      producto = a * b * c
      IMPRIMIR "El producto es:",producto
FIN SI
```

SINO es opcional por lo que la instrucción puede escribirse:

Sintaxis:

```
SI condición ENTONCES

      instrucción(es)

FIN SI
```

Ejemplo:

```
INGRESAR a
SI a > b ENTONCES
      suma = a + 10
      IMPRIMIR "La suma es:",suma
FIN SI
```

Los siguientes ejercicios resueltos nos darán una mayor comprersión sobre la utilización de este tipo de estructura.

Ejercicio 3

Dado un número, decir si es positivo, negativo o cero.


```
COMIENZO
INGRESAR "Ingrese un número:", num
SI num > 0 ENTONCES
          IMPRIMIR "número positivo"
SINO
          SI num < 0 ENTONCES
                    IMPRIMIR "número negativo"
          SINO
                    IMPRIMIR " número igual a cero"
          FIN SI
FIN SI
FIN
```

Se puede apreciar el anidamiento de dos instrucciones SI ENTONCES SINO: una instrucción dentro de otra.

Ejercicio 4

Dados dos números, calcular:

a) La suma, si el primero es mayor que el segundo.
b) La diferencia, si el primero es menor que el segundo.
c) El producto, si son iguales.

```
COMIENZO
INGRESAR "Ingrese dos números:", num1, num2
SI num1 > num2 ENTONCES
               suma = num1 + num2
               IMPRIMIR " La suma es:", suma
SINO
               SI num2 > num1 ENTONCES
                            resta = num2 – num1
                            IMPRIMIR "La resta es:", resta
               SINO
                            prod = num1 * num2
                            IMPRIMIR "El producto es:", prod
               FIN SI
FIN SI
FIN
```

Nuevamente al igual que el ejercicio anterior, se anidan dos decisiones para poder evaluar las dos condiciones determinadas por el enunciado del problema.

Ejercicio 5

Dados tres lados de un triángulo, decir que tipo de triángulo es.

```
COMIENZO
INGRESAR "Ingrese la longitud de los lados:", a, b, c
SI a = b AND b = c ENTONCES
                IMPRIMIR "Triángulo equilatero"
SINO
                SI a <> b AND b <> c ENTONCES
                                IMPRIMIR "Triángulo escaleno"
                SINO
                                IMPRIMIR "Triángulo isósceles"
                FIN SI
FIN SI
FIN
```

En este ejercicio se recurre al operador lógico AND, para ampliar la capacidad de pregunta en las condiciones evaluadas por la instrucción de decisión.

Así, al colocar a = b AND b = c , puedo hacer una doble pregunta sin tener que escribir dos instrucciones de decisión.

Ejercicio 6

Dados dos números X y Z, sumarlos si X es mayor que Z y restarlos en caso contrario.

```
COMIENZO
INGRESAR "Ingrese dos números:", x, z
SI x > z ENTONCES
        res = x + z
SINO
```



```
        SI x < z ENTONCES res = z - x
FIN SI
IMPRIMIR "El resultado es:", res
FIN
```

Este ejercicio permite apreciar, características comunes en el planteo de la estructura: ingreso de los datos, evaluación de las condiciones e impresión de resultados.

Ejercicio 7

Dados tres números, se pide:

a) La suma.
b) El promedio.
c) Si el promedio es mayor que 5, imprimir un mensaje que diga "El promedio es mayor que 5".

```
COMIENZO
INGRESAR "Ingrese tres números:", a, b, c
suma = a + b + c
prom = suma / 3
IMPRIMIR "La suma es:", suma
IMPRIMIR "El promedio es:", prom
SI prom > 5 ENTONCES IMPRIMIR "El promedio es mayor que 5"
FIN
```

La instrucción de decisión se utiliza en este caso, para imprimir un mensaje, que estará condicionado por el contenido de una de las variables numéricas calculadas (prom).

Como variante de resolución, se podría haber impreso directamente el cálculo sobre la instrucción de impresión. Observemos como quedaría:

```
COMIENZO
INGRESAR "Ingrese tres números:", a, b, c
IMPRIMIR "La suma es:", a + b + c
IMPRIMIR "El promedio es:", suma / 3
SI prom > 5 ENTONCES IMPRIMIR "El promedio es mayor que 5"
FIN
```

Ejercicio 8

Dados 4 números, decir si la suma de los dos primeros es mayor a la suma de los dos segundos.

```
COMIENZO
INGRESAR "Ingrese cuatro números:", a, b, c, d
SI (a + b) > (c + d) ENTONCES
            IMPRIMIR "La suma de los dos primeros números es mayor"
SINO
          IMPRIMIR "La suma de los dos primeros números no es mayor"
FIN SI
FIN
```

Se puede ver como en lugar de evaluar el contenido de dos variables, se comparan dos expresiones: a + b con c + d.

Ejercicio 9

Se conocen las edades y estaturas de 3 alumnos de un curso. Se pide:

a) Imprimir la edad promedio.
b) Imprimir la estatura promedio.
c) Imprimir las edades de los alumnos mayores de 15 años, que miden menos de 1.5 metros.

```
COMIENZO
INGRESAR "Ingrese las 3 edades:", edad1, edad2, edad3
INGRESAR "Ingrese las 3 alturas:", altura1, altura2, altura3
edadprom = (edad1 + edad2 + edad3) / 3
altuprom = (altura1 + altura2 + altura3) / 3
IMPRIMIR "La edad promedio es:", edadprom
IMPRIMIR "La altura promedio es:", altuprom
SI edad1 > 15 AND altura1 < 1.5 ENTONCES IMPRIMIR edad1
SI edad2 > 15 AND altura2 < 1.5 ENTONCES IMPRIMIR edad2
SI edad3 > 15 AND altura3 < 1.5 ENTONCES IMPRIMIR edad3
FIN
```


Nuevamente al igual que en el ejercicio 5, se vuelve a utilizar el operador lógico AND, esta vez para condicionar doblemente la impresión de los resultados .

Ejercicio 10

Una empresa paga sueldos calculando el valor de la hora y la cantidad de horas que trabajó cada empleado. Además si el empleado trabajó más de 100 horas, lo premian con 100$ y si trabajó más de 200 horas, le dan 50$ más.
Hallar el sueldo del empleado.

```
COMIENZO
INGRESAR "Ingrese el valor de la hora:", vh
INGRESAR "Ingrese la cantidad de horas trabajadas:", ht
basico = vh * ht
SI ht > 200 ENTONCES
          sueldo = basico + 150
SINO
          SI ht > 100 ENTONCES
                    sueldo = basico + 100
          SINO
                    sueldo = basico
          FIN SI
FIN SI
FIN
```

En este caso se ingresaron dos variables, vh y ht, cuyo producto se asignará a una nueva variable: basico. Esta, más la suma del valor adicional según corresponda, se asignará ahora a la variable sueldo para su posterior impresión.

Ejercicio 11

Dadas la cantidad de horas trabajadas, la categoría y la antigüedad de un empleado, calcular el sueldo teniendo en cuenta que cobra 50$ adicionales por cada año trabajado.
El valor de la hora para cada categoría es:

```
Categoría 1: 10$
Categoría 2: 15$
Categoría 3: 20$

COMIENZO
INGRESAR "Horas trabajadas:", ht
INGRESAR "Categoría:", cat
INGRESAR "Antigüedad:", ant
SI cat = 1 ENTONCES
          sueldo = ht * 10 + 50 * ant
SINO
          SI cat = 2 ENTONCES
                    sueldo = ht * 15 + 50 * ant
          SINO
                    sueldo = ht * 20 + 50 * ant
          FIN SI
FIN SI
FIN
```

Estamos en presencia nuevamente de decisiones anidadas, esto es así porque se evalúan los valores de cat que condiciona el cálculo de los distintos sueldos.

2.2.4 Instrucción de decisión múltiple: SELECCIONAR CASO.......

Esta instrucción permite la selección de un conjunto de acciones a partir de una lista de diferentes opciones o casos.

La instrucción de decisión múltiple facilita la selección de más de dos alternativas, haciendo los programas más flexibles y más fáciles de leer y de escribir.

Sintaxis:

```
SELECCIONAR CASO  variable
     CASO condición1
          instrucción(es)
     CASO condición2
            instrucción(es)
     CASO condición3
```



```
            instrucción(es)
            ...............
            ...............
     OTRO CASO
            instrucción(es)
FIN SELECCIONAR
```

SELECCIONAR CASO proporciona el resultado que permite la ejecución de un bloque de sentencias CASO, pues cada secuencia de instrucciones evalúa diferentes resultados de la variable.

La alternativa "opcional" OTRO CASO proporciona un bloque de instrucciones a ejecutarse, sólo en el caso que todos los resultados evaluados sean falsos.

Ejemplo:

```
SELECCIONAR CASO numero
     CASO < 0
            IMPRIMIR "Número negativo"
     CASO > 0
            IMPRIMIR "Número positivo"
     OTRO CASO
            IMPRIMIR "Número igual a cero"
FIN SELECCIONAR
```

Se puede apreciar la ventaja que presenta la decisión multiple en el análisis del siguiente caso.

Si volvemos al ejercicio 11 pero agregando dos categorías más, observaremos como se dificulta la escritura del programa afectando su legibilidad, pues deberíamos colocar al menos dos decisiones simples más.

Se produce un anidamiento de las decisiones, lo cual en el caso de ser muchas las instrucciones de decisión "anidadas", es poco práctico.

Ahora si en lugar de esto, utilizamos una decisión múltiple veremos como la resolución del ejercicio, gana en claridad.

A continuación resolveremos el ejercicio 12 de las dos maneras: sin y con decisión múltiple.

Ejercicio 12

Dadas la cantidad de horas trabajadas, la categoría y la antigüedad de un empleado, calcular el sueldo teniendo en cuenta que cobra 50$ adiciona-

les por cada año trabajado.

El valor de la hora para cada categoría es:

Categoría 1: 10$
Categoría 2: 12$
Categoría 3: 15$
Categoría 4: 18$
Categoría 5: 20$

- Caso a: resolución " anidando" SI.. ENTONCES..SINO

```
COMIENZO
INGRESAR "Horas trabajadas:", ht
INGRESAR "Categoría:", cat
INGRESAR "Antigüedad:", ant
SI cat = 1 ENTONCES
          sueldo = ht * 10 + 50 * ant
SINO
          SI cat = 2 ENTONCES
                    sueldo = ht * 12 + 50 * ant
          SINO
                    SI cat = 3 ENTONCES
                              sueldo = ht * 15 + 50 * ant
                    SINO
                              SI cat = 4 ENTONCES
                                        sueldo = ht * 18 + 50 * ant
                              SINO
                                        sueldo = ht * 20 + 50 * ant
                              FIN SI
                    FIN SI
          FIN SI
FIN SI
FIN
```

- Caso b: resolución utilizando SELECCIONAR CASO

```
COMIENZO
INGRESAR "Horas trabajadas:", ht
INGRESAR "Categoría:", cat
INGRESAR "Antigüedad:", ant
SELECCIONAR CASO cat
```



```
             CASO = 1
                    vh = 10
             CASO = 2
                    vh = 12
             CASO = 3
                    vh = 15
             CASO = 4
                    vh = 18
             CASO = 5
                    vh = 20
FIN SELECCIONAR
sueldo = ht * vh
IMPRIMIR "El sueldo es:", sueldo
FIN
```

RESUMEN

Las instrucciones de decisión, realizan acciones alternativas, pues la ejecución de una instrucción o grupo de instrucciones depende de si se cumple o no, una o varias condiciones.

Existen distintos tipos de formato para instrucciones de decisión:

- **Instrucción de decisión simple:SI......ENTONCES......**

Sintaxis: SI condición ENTONCES instrucción

- **Instrucción de decisión doble: SI......ENTONCES.......SINO.......**

Sintaxis: SI condición ENTONCES instrucción1 SINO instrucción2

- **Instrucción de decisión en bloques:**

Sintaxis:

```
SI condición ENTONCES
      instrucción(es)
SINO
      instrucción(es)
FIN SI
```

SINO es opcional por lo que la instrucción puede escribirse:

Sintaxis:

```
SI condición ENTONCES
      instrucción(es)
FIN SI
```


- **Instrucción de decisión múltiple: SELECCIONAR CASO........**

Sintaxis:

```
SELECCIONAR CASO variable
      CASO condición1
            instrucción(es)
      CASO condición2
            instrucción(es)
      CASO condición3
            instrucción(es)
            ................
            ................
      OTRO CASO
            instrucción(es)
FIN SELECCIONAR
```



Ejercicios Propuestos



1. Dados dos números a y b; sumarlos si a >= b y restarlos si a < b.

2. Dados dos números a y b, calcular el cociente a / b. Considerar que si b es cero, debe aparecer un mensaje aclaratorio que indique operación no definida.

3. Dados dos números, imprimirlos ordenados de mayor a menor.

4. Se conocen las notas de historia y geografía de 3 alumnos de un curso. Se pide:

a) Imprimir el promedio de notas de historia.
b) Imprimir el promedio de notas de geografía.
c) Imprimir el promedio total de cada alumno.

5. Dados la cantidad de horas trabajadas y la categoría de un empleado, calcular el sueldo de bolsillo teniendo en cuenta que los descuentos totales son del 20 %.
El valor de la hora para cada categoría es:

Categoría 1: 12$
Categoría 2: 15$
Categoría 3: 18$
Categoría 4: 20$
Categoría 5: 25$
Categoría 6: 28$
Categoría 7: 30$

capítulo 3

Estructuras de Repetición

3.1 INTRODUCCIÓN

Este tipo de estructuras nos dan la posibilidad de repetir un conjunto de instrucciones un número determinado de veces .
De acuerdo a la forma de control de la cantidad de repeticiones podemos diferenciar dos grandes grupos de instrucciones:

- controladas por condición.
- controladas por contador.

En el transcurso de este capítulo analizaremos las distintas variantes referidas a este tipo de estructuras. Además introduciremos conceptos de connotaciones eminentemente prácticas como los referidos a la utilización de contadores y acumuladores.
Sobre el final presentaremos el concepto de máximos y mínimos, a partir de algunas aplicaciones.

3.2 INSTRUCCIONES CONTROLADAS POR CONDICIÓN:

3.2.1 Instrucción HACER MIENTRAS

La computadora examina la condición ,si se cumple ejecuta las instrucciones que están dentro del ciclo hasta que deje de cumplirse. Entonces sale del ciclo y continua ejecutando las instrucciones que están a continuación .
Si no se cumple la condición no entra en el ciclo.

Sintaxis:

```
HACER MIENTRAS condición
    instruccion(es)
REPETIR
```

Ejemplo:

```
INGRESAR a
HACER MIENTRAS  a > 0
      IMPRIMIR a
INGRESAR a
REPETIR
```

3.2.2 Instrucción HACER HASTA

Es similar a HACER MIENTRAS pero evalúa en forma inversa la condición de tal manera que ejecuta las instrucciones hasta que deje de cumplirse .

Sintaxis:

```
HACER
      instruccion(es)
 REPETIR HASTA condición
```

Ejemplo:

```
      INGRESAR a
HACER
      IMPRIMIR a
      INGRESAR a
REPETIR HASTA a < 0
```

Se puede apreciar en el ejemplo que la estructura HACER HASTA ,asegura que por lo menos una vez se va a ejecutar el bloque de instrucciones dentro del ciclo ,lo que podía no suceder en una estructura HACER MIENTRAS.

3.2.3 Ejercicios Resueltos

Para comprender mejor el funcionamiento de las instrucciones de repetición, controlados por condición, analizaremos los siguientes ejercicios:

Ejercicio 1

Dados un conjunto de números, imprimir los números que sean mayores o iguales a 5, suponiendo que se ingresan números hasta uno igual a cero.


```
COMIENZO
INGRESAR "Ingrese un número:", num
HACER MIENTRAS num <> 0
          SI num >= 5 ENTONCES IMPRIMIR num
          INGRESAR "Ingrese un número:", num
REPETIR
FIN
```

En el ejercicio 1 se observa como se debe colocar dos veces la instrucción INGRESAR, la primera para poder ingresar por primera vez en el ciclo a partir de la evaluación de la variable num y la segunda para volver a entrar luego de cada repetición.

Este concepto se reitera en todos los ejercicios que utilicen estructuras repetitivas.

La impresión del cotenido de la variable numérica num, está condicionada por el valor de la misma variable: mayor o igual que 5.

Si resolvemos el mismo ejercicio utilizando la instrucción HACER HASTA, está asegurado que el ciclo se ejecutará al menos una vez, no siendo necesario repetir dos veces el ingreso en la estructura.

```
COMIENZO
HACER
          INGRESAR "Ingrese un número:", num
          SI num >= 5 ENTONCES IMPRIMIR num
REPETIR HASTA num <> 0
FIN
```

Ejercicio 2

Dadas las notas y nombres de alumnos de un curso, imprimir los nombres de los alumnos, cuya nota sea mayor que 7. El ingreso finaliza con nota igual a cero.

```
COMIENZO
INGRESAR "Ingrese el nombre y la nota:", nom$, nota
HACER MIENTRAS nota <> 0
          SI nota > 7 ENTONCES IMPRIMIR nom$
          INGRESAR "Ingrese el nombre y la nota:", nom$, nota
REPETIR
FN
```

El ejercicio es muy similar al anterior en su estructura. La diferencia radica en que ahora la impresión de la variable, alfanumérica en este caso, se encuentra condicionada por la variable numérica.

Utilizando HACER HASTA:

```
COMIENZO
HACER
            INGRESAR "Ingrese el nombre y la nota:", nom$, nota
            SI nota > 7 ENTONCES IMPRIMIR nom$
REPETIR HASTA nota <> 0
FIN
```

3.3 INSTRUCCIONES CONTROLADAS POR CONTADOR:

3.3.1 Instrucción PARA...PRÓXIMO

Se utiliza cuando conozco exactamente la cantidad de veces que se tiene que repetir la ejecución del bloque de instrucciones que conforman la estructura repetitiva.

```
Sintaxis:     PARA variable indice = valor inicial  A valor final  PASO
incremento

                        instruccion(es)

              PROXIMO
```

El bloque de instrucciones comienza a repetirse de acuerdo al contador que está definido por el valor inicial de arranque que toma la variable índice, el valor final de parada y un PASO que indica el incremento fijo que tiene en cada vuelta de repetición dicha variable índice.

Cuando el incremento del PASO es uno , su escritura es opcional.

```
Ejemplo:        PARA  contador = 1 A 50  PASO 1
                      INGRESAR "Número:",numero
                      IMPRIMIR "Número ingresado:",numero
                PROXIMO
```


3.3.2 Ejercicios Resueltos

A continuación analizaremos algunos ejercicios donde podremos apreciar el funcionamiento de la instrucción PARA...PROXIMO

Ejercicio 3

Dados 25 números, imprimir los números que sean mayores o iguales a 5.

```
COMIENZO
PARA c = 1 A 25
          INGRESAR "Ingrese un número:", num
          SI num >=  5 ENTONCES IMPRIMIR num
PROXIMO
FIN
```

En este caso se omitió colocar el paso, por ser igual a uno. En los ejercicios que analicemos a continuación, como norma, no colocaremos el paso cada vez que el incremento sea uno.

La variable c, es la encargada de contar el número de repeticiones.

La impresión del número está condicionada a que sea mayor o igual a cinco.

Ejercicio 4

Dadas 15 notas y nombres de alumnos de un curso, imprimir los nombres de los alumnos cuya nota sea mayor a 7.

```
COMENZO
PARA c = 1 A 15
          INGRESAR "Ingrese nombre y nota:", nom$, nota
          SI nota >=  7 ENTONCES IMPRIMIR nom$
PROXIMO
FIN
```

La estructura de este ejercicio es muy similar al anterior, a diferencia que ahora la variable numérica nota, condiciona la impresión de la variable alfanumérica nom$.

Ejercicio 5

Dados n notas y nombres de alumnos de un curso, imprimir los nombres de los alumnos cuya nota sea mayor que 7.

```
COMIENZO
INGRESAR "Ingrese la cantidad de alumnos:", n
PARA c = 1 A n
        INGRESAR "Ingrese nombre y nota:", nom$, nota
        SI nota >= 7 ENTONCES IMPRIMIR nom$
PROXIMO
FIN
```

El ejercicio 5 es una variante del 4. La cantidad de repeticiones que tendrá el ciclo PARA, no está explicitada en el enunciado, sino que debe ingresarse en la variable n, desde afuera. Esto otorga un alto grado de flexibilidad en la manipulación de la cantidad de datos a procesar.

Ejercicio 6

Dados 10 números, imprimir para cada uno si es positivo o negativo.

```
COMIENZO
PARA c = 1 A 10
        INGRESAR "Ingrese un número:", num
        SI num > 0 ENTONCES
                IMPRIMIR "Positivo"
        SINO
                SI num < 0 ENTONCES IMPRIMIR "Negativo"
        FIN SI
PROXIMO
FIN
```

Se observa en el ejercicio 6 trabajar juntas a las estructuras de control: decisión y repetición.

3.4 UTILIZACIÓN DE VARIABLES COMO ACUMULADOR

Un acumulador es una variable que se incrementa o decrementa en un **valor variable**.

Sintaxis: variable acumulador = variable acumulador +(-) variable

Ejemplo: saldo = saldo + valor

La computadora suma el contenido de la variable valor , a la variable saldo y asigna el resultado a la variable saldo de la izquierda.

3.4.1 Ejercicios Resueltos

Veamos algunos ejercicios de utilización de acumuladores:

Ejercicio 7

Dados 15 números, imprimir la suma total.

```
COMIENZO
suma = 0
PARA c = 1 A 15
        INGRESAR "Ingrese un número:", num
        suma = suma + num
PROXIMO
IMPRIMIR "La suma total es:", suma
FIN
```

En este ejercicio se utilizó como acumulador a la variable suma, a la que previamente se le asignó cero, para evitar arrastrar algún valor aleatorio que pueda tener almacenado.

Es una buena práctica asignar en todos los casos, un valor inicial (inicializar) a las variables que se utilicen como acumuladores.

Ejercicio 8

Dados números hasta ingresar uno negativo, imprimir la suma total.

```
COMIENZO
suma = 0
INGRESAR "Ingrese un número:", num
HACER MIENTRAS num < 0
        suma = suma + num
        INGRESAR "Ingrese un número:", num
REPETIR
IMPRIMIR "La suma total es:', suma
FIN
```

Este ejercicio presenta la variante con respecto al 7, en la cantidad de valores a procesar la cual es desconocida. Como se sabe que el ingreso finaliza con un número igual a cero, se utilizó una instrucción HACER MIENTRAS para controlar el ingreso de los datos a procesar.

Ejercicio 9

Dados n números, imprimir el promedio.

```
COMIENZO
INGRESAR "Ingrese la cantidad de números:", n
suma = 0
PARA c = 1 A n
        INGRESAR "Ingrese un número:", num
        suma = suma + num
PROXIMO
prom = suma / n
IMPRIMIR "El promedio es:', prom
FIN
```

En este caso la cantidad de números a procesar es variable pero siempre conocida al momento de comenzar dicho proceso.

Como ahora se pide la impresión del promedio, este se obtiene de dividir el contenido acumulado en sum, por la cantidad de valores procesados (n).

3.5 UTILIZACIÓN DE VARIABLES COMO CONTADOR DE EVENTOS

Un contador es una variable que se incrementa o decrementa en un **valor constante**.

Sintaxis: variable contador = variable contador +(-) constante

Ejemplo: contador = contador - 1

De manera similar a la variable acumulador, el contador debe ser previamente cargado con un valor inicial a partir de cual comenzará a contar.

3.5.1 Ejercicios Resueltos

Ejercicio 10

Dados 10 números, imprimir cuántos son positivos, cuantos son negativos, y cuantos son cero.

```
COMIENZO
pos = 0
neg = 0
PARA i = 1 A 10
        INGRESAR "Ingrese un número:", nro
SI nro > 0 ENTONCES
        pos = pos + 1
SINO
        SI nro < 0 ENTONCES neg = neg + 1
FIN SI
PROXIMO
ceros = 10 - pos - neg
IMPRIMIR 'Cantidad de positivos:", pos
IMPRIMIR 'Cantidad de negativos:", neg
IMPRIMIR 'Cantidad de ceros:", ceros
FIN
```

Se trabajo con dos contadores: uno para los números positivos y otro para los negativos. En el caso de la cantidad de ceros se podría haber utilizado otro contador, pero en este caso, se optó por calcularlo por diferencia

Ejercicio 11

Dados los sueldos de n empleados, imprimir cuantos ganan más de 2000$ y cuantos ganan menos de esa suma.

```
COMIENZO
mas = 0
menos = 0
INGRESAR "Ingrese la cantidad de sueldos:", n
PARA i = 1 A n
        INGRESAR "Ingrese el sueldo:", sdo
        SI sdo > 2000 ENTONCES
                mas = mas + 1
        SINO
                SI sdo < 2000 ENTONCES  menos = menos + 1
        FIN SI
PROXIMO
IMPRIMIR "Cantidad de sueldos mayores a 2000$:", mas
IMPRIMIR "Cantidad de sueldos menores a 2000$:", menos
FIN
```

Al igual que en el ejercicio 10, se utilizaron dos contadores. Esta vez uno para los sueldos mayores a 2000$ (mas) y otro para los sueldos menores a 2000$.

Ejercicio 12

Dadas 20 notas de alumnos de un curso, imprimir:

a) Cantidad de alumnos aprobados (3 < nota < 7).

b) Cantidad de alumnos aplazados (nota < 4).

c) Cantidad de alumnos promocionados (nota > 6).


```
COMIENZO
apr = 0
apl = 0
pro = 0
PARA i = 1 A 20
        INGRESAR "Ingrese la nota:", nota
        SI nota > 6 ENTONCES
                pro = pro - 1
        SINO
                SI nota < 4 ENTONCES
                        apl = apl + 1
                SINO
                        apr = apr + 1
                FIN SI
        FIN SI
PROXIMO
IMPRIMIR "Cantidad de promocionados:", pro
IMPRIMIR "Cantidad de aplazados:", apl
IMPRIMIR "Cantidad de aprobados:", apr
FIN
```

En este caso se utilizaron tres contadores, los que fueron previamente inicializados en cero.

Además de tener una decisión dentro del ciclo de repetición PARA, encontramos una segunda instrucción de decisión anidada dentro de la primera.

3.7 MÁXIMOS Y MÍNIMOS

Una de las aplicaciones clásicas en programación, es la obtención de un valor máximo o de un valor mínimo entre un conjunto de valores.

Las estructuras de decisión y repetición estudiadas, nos brindan las herramientas necesarias para el desarrollo de los algoritmos que constituyan la respuesta a nuestro problema.

Los ejercicios que se analizan a continuación, nos permiten visualizar los mecanismos de resolución.

En todos los ejemplos se consideró como restricción, que sólo existe un máximo o un mínimo. En unidades posteriores analizaremos la posibilidad que exista más de un máximo o de un mínimo.

Ejercicio 13

Dadas dos temperaturas, imprimir la mayor (consideramos que las temperaturas son distintas).

```
COMIENZO
INGRESAR "Ingrese una temperatura:", temp1
INGRESAR "Ingrese otra temperatura:", temp2
SI temp1 > temp2 ENTONCES IMPRIMIR temp1 SINO IMPRIMIR temp2
FIN
```

En el ejemplo se compararon dos temperaturas y se imprimió la mayor.

Ejercicio 14

Dados 15 sueldos, imprimir el máximo.

```
COMIENZO
max = 0
PARA i = 1 A 15
          INGRESAR "Ingrese el sueldo:", sdo
          SI sdo > max ENTONCES max = sdo
PROXIMO
IMPRIMIR "El sueldo máximo es:", max,"$"
FIN
```

Se utilizó una variable max, para almacenar el sueldo máximo. Esta variable se inicializó en cero, para garantizar que en la primera pasada por el ciclo PARA, el primer sueldo sea considerado como máximo. Sólo cuando se ingrese en una futura pasada un valor mayor de sueldo, se reasignará nuevamente max.

Al finalizar la ejecución del ciclo PARA, quedará almacenado en max el valor del sueldo máximo .

En realidad se podría haber inicializado max con cualquier valor que asegure ser menor que los valores de sueldo que se van a procesar.

Otra forma sería la siguiente:


```
COMIENZO
INGRESAR "Ingrese el sueldo:", sdo
max = sdo
PARA i = 2 A 15
          INGRESAR "Ingrese el sueldo:", sdo
          SI sdo > max ENTONCES max = sdo
PROXIMO
IMPRIMIR "El sueldo máximo es:", max, "$"
FIN
```

En este caso se ingresa el primer sueldo y se lo asigna directamente a la variable max, luego se arranca con el ciclo PARA a partir del segundo valor de sueldo (i = 2).

Ejercicio 15

Dados 15 sueldos, imprimir el mínimo.

```
COMIENZO
min = 9999999999
PARA i = 1 A 15
          INGRESAR "Ingrese el sueldo:", sdo
          SI sdo < min ENTONCES min = sdo
PROXIMO
IMPRIMIR "El sueldo mínimo es:", min,"$"
FIN
```

Se puede apreciar que el algoritmo es muy similar al cálculo del máximo. Se inicializó la variable min con, un valor cualquiera tan alto, que garantice que en la primera pasada por el ciclo PARA, se almacene en min el primer sueldo. De la misma manera que en la determinación del máximo, se reasignará min cuando se ingrese un valor inferior al que fue previamente almacenado.

Al finalizar el ciclo, min guardará el menor valor de sueldo de los 15 leidos.

Otra resolución del ejercicio igualmente correcta sería

```
COMIENZO
INGRESAR "Ingrese el sueldo:", sdo
min = sdo
PARA i = 2 A 15
        INGRESAR "Ingrese el sueldo:", sdo
        SI sdo < min ENTONCES min = sdo
PROXIMO
IMPRIMIR "El sueldo mínimo es:", min,"$"
FIN
```

En este caso se asigna directamente el primer sueldo a min fuera del ciclo PARA, para poder comparar luego dentro del ciclo con los catorce valores restantes.

Ejercicio 16

Dadas 15 edades y nombres de alumnos de un curso, imprimir el nombre del alumno de edad máxima.

```
COMIENZO
max = 0
PARA i = 1 A 15
        INGRESAR "Ingrese el nombre del alumno:", nombre$
        INGRESAR "Ingrese la edad del alumno:", edad
        SI edad > max ENTONCES
                max = edad
                nomax$ = nombre$
        FIN SI
PROXIMO
IMPRIMIR "El nombre del alumno de mayor edad es:", nomax$
FIN
```

Observamos como se determina el valor máximo, en el ejemplo una variable numérica, pero se imprime el nombre asignado a una variable alfanumérica, que corresponde a ese valor máximo.

Ejercicio 17

En una carrera de autos compiten 45 autos. Al finalizar las vueltas de clasificación se tienen los tiempos de cada auto.

Se pide imprimir:

a) el número de auto que llegó primero.

b) el número de auto que llegó último.

```
COMIENZO
tmax = 0
tmin = 9999999999
PARA i = 1 A 45
        INGRESAR "Ingrese el número de auto:", auto
        INGRESAR "Ingrese el tiempo del auto:", tiempo
        SI tiempo > max ENTONCES
                tmax = tiempo
                ultimo = auto
        FIN SI
        SI tiempo < min ENTONCES
                tmin = tiempo
                primero = auto
        FIN SI
PROXIMO
IMPRIMIR "El auto de mejor tiempo es:", primero
IMPRIMIR "El auto de peor tiempo es:", ultimo
FIN
```

En este ejercicio se combina la determinación de un máximo (variables tmax y ultimo) con la de un mínimo (variables tmin y primero).
Ambos procesos forman parte de un mismo ciclo PARA.

RESUMEN

Este tipo de estructuras nos dan la posibilidad de repetir un conjunto de instrucciones un número determinado de veces.

De acuerdo a la forma de control de la cantidad de repeticiones podemos diferenciar dos grandes grupos de instrucciones: controladas por condición y controladas por contador.

Instrucciones controladas por condición:

✓ Instrucción HACER MIENTRAS

La computadora examina la condición ,si se cumple ejecuta las instrucciones que están dentro del ciclo hasta que deje de cumplirse. Entonces sale del ciclo y continua ejecutando las instrucciones que están a continuación .

Si no se cumple la condición no entra en el ciclo.

Sintaxis:

```
HACER MIENTRAS condición
    instruccion(es)
REPETIR
```

✓ Instrucción HACER HASTA

Es similar a HACER MIENTRAS pero evalúa en forma inversa la condición de tal manera que ejecuta las instrucciones hasta que deje de cumplirse .

Sintaxis:

```
HACER
    instruccion(es)
REPETIR HASTA condición
```

La estructura HACER HASTA ,asegura que por lo menos una vez se va a ejecutar el bloque de instrucciones dentro del ciclo ,lo que podía no suceder en una estructura HACER MIENTRAS.

Instrucciones controladas por contador:

✓ Instrucciones PARA...PRÓXIMO

Se utiliza cuando conozco exactamente la cantidad de veces que se tiene que repetir la ejecución del bloque de instrucciones que conforman la estructura repetitiva.

Sintaxis: PARA variable indice = valor inicial A valor final PASO incremento

```
        instruccion(es)
PROXIMO
```

El bloque de instrucciones comienza a repetirse de acuerdo al contador que está definido por el valor inicial de arranque que toma la variable índice, el valor final de parada y un PASO que indica el incremento fijo que tiene en cada vuelta de repetición dicha variable índice.

Cuando el incremento del PASO es uno , su escritura es opcional.

✓ Acumulador

Un acumulador es una variable que se incrementa o decrementa en un **valor variable**.

Sintaxis: variable acumulador = variable acumulador +(-) variable

Es una buena práctica asignar en todos los casos, un valor inicial (inicializar) a las variables que se utilicen como acumuladores, para evitar arrastrar algún valor aleatorio que pueda tener almacenado.

✓ Contador

Un contador es una variable que se incrementa o decrementa en un **valor constante**.

Sintaxis: variable contador = variable contador +(-) constante

De manera similar a la variable acumulador, el contador debe ser previa-

mente cargado con un valor inicial a partir del cual comenzará a contar.

✓ Máximos y Mínimos

Una de las aplicaciones clásicas en programación, es la obtención de un valor máximo o de un valor mínimo entre un conjunto de valores.

Las estructuras de decisión y repetición estudiadas, nos brindan las herramientas necesarias para el desarrollo de los algoritmos que constituyan la respuesta a nuestro problema.



Ejercicios Propuestos

1. Dados 50 sueldos, mprimir:

a) la suma.

b) cantidad de sueldos mayores que 1500$.

2. Dadas las edades y estaturas de 45 alumros de un curso, se pide:

a) edad promedio.

b) estatura promedio.

c) cantidad de alumnos mayores de 10 años.

d) cantidad de alumnos que miden menos de 1.40 mts.

3. Una empresa elaboradora de desinfectantes incrementa las ventes debido al cólera.

En cada factura de venta se registra la siguiente información

-Número de factura.
-Número de artículo.
-Cantidad en litros
-Precio unitario por litro.

El ingreso de datos finaliza con número de factura igual a cero.
Se pide:

a) Facturación mensual.

b) ¿Cuántos litros se vendieron del artículo 1?

c) ¿Cuántas facturas de más de 300$ se emitieron?

4. En una Universidad los alumnos tienen una nota que resulta de sacar el promedio de todas las materias. Hay 700 alumnos.
Se pide:

a) la cantidad de alumnos con nota promedio superior a 6.

b) si la cantidad de alumnos con nota promedio inferior a 4 es superior a 500, imprimir un mensaje que diga "Esta Universidad tiene un promedio muy bajo".

5. Dados n sueldos, imprimir el máximo.

6. Dados n nombres de artículos y sus correspondientes precios, imprimir:

a) el nombre del artículo más caro.

b) el precio del artículo más barato.

Ejercicios de Propósito General

4.1 INTRODUCCIÓN

En este capítulo desarrollaremos una serie de ejercicios que resumen todos los conceptos analizados hasta ahora.
El objetivo principal es aprender a razonar las soluciones desde el punto de vista informático, utilizando las herramientas descriptas en los capítulos anteriores.
Para lograr lo expresado anteriormente, trataremos de combinar los conocimientos adquiridos y así poder integrarlos en la resolución de los ejercicios planteados.

4.2 EJERCICIOS RESUELTOS

Ejercicio 1

En una empresa los empleados cobran un sueldo según la categoría a la que pertenecen. Son 50 empleados.
Los sueldos son:

Categoría 1.............1500$
Categoría 2.............1700$
Categoría 3.............2000$

Al sueldo se le suma además 100$ por cada año trabajado.
Si para cada empleado se conoce su categoría y antigüedad, se pide calcular e imprimir:

a) La cantidad de empleados por categoría.
b) El total de sueldos pagados por categoría.
c) El sueldo promedio.

d) El sueldo máximo y la categoría a la que pertenece.

```
COMIENZO
ce1 = 0: ce2 = 0: ce3 = 0
ts1 = 0: ts2 = 0: ts3  = 0
smax = 0
PARA empleado = 1 A 50
        INGRESAR "Categoría:", cat
        INGRESAR "Antigüedad:", ant
        SELECCIONAR CASO cat
                CASO = 1
                        sdo = 1500 + ant * 100
                        ts1 = ts1 + sdo
                        ce1 = ce1 + 1
                CASO = 2
                        sdo = 1700 + ant * 100
                        ts2 = ts2 + sdo
                        ce2 = ce2 + 1
                CASO = 3
                        sdo = 2000 + ant * 100
                        ts3 = ts3 + sdo
                        ce3 = ce3 + 1
        FIN SELECCIONAR
        SI sdo > smax ENTONCES
                smax = sdo
                cmax = cat
        FIN SI
 PROXIMO
 sprom = (ts1 + ts2 + ts3) / 50
 IMPRIMIR "Cantidad de empleados de la categoría 1:", ce1
 IMPRIMIR "Cantidad de empleados de la categoría 2:", ce2
 IMPRIMIR "Cantidad de empleados de la categoría 3:", ce3
 IMPRIMIR "Total de sueldos de la categoría 1:", ts1, "$"
 IMPRIMIR "Total de sueldos de la categoría 2:", ts2, "$"
 IMPRIMIR "Total de sueldos de la categoría 3:", ts3, "$"
 IMPRIMIR "Sueldo promedio:", sprom, "$"
 IMPRIMIR "El sueldo máximo es:", smax, "$  y corresponde a la catego-
ría:", cmax
 FIN
```


Lo primero que debemos hacer cuando comenzamos a escribir un programa, es inicializar todas las variables que vamos a utilizar como: contadores, acumuladores, máximos y mínimos.

Para la resolución se utilizaron tres contadores para la cantidad de empleados, ce1, ce2 y ce3, y tres acumuladores de sueldo por categoría, ts1, ts2 y ts3, los que previamente fueron iniciados a cero. **Por una cuestión práctica con respecto a la escritura, se separaron estas asignaciones iniciales por dos puntos (:), para no tener que cambiar permanentemente de línea de instrucción.** Esto último es válido en la mayoría de los lenguajes de programación.

La variable empleado, se utilizó como variable de conteo del ciclo PARA y va de 1 a 50.

En la entrada de datos se ingresa la categoría y la antigüedad. Luego se evalúa a partir de una selección múltiple, la categoría para determinar que variables tendrán en cuenta los datos leidos, para el cálculo del total de sueldos y la cantidad de empleados por categoría.

A continuación se determinó el valor del sueldo máximo (smax) y de la categoría a la que pertenece (cmax). **Con respecto a la variable cmax, no fue necesario colocarla en cero en el comienzo, pues en realidad no almacena un valor máximo, sino la categoría a la cual pertenece dicho valor.**

La variable sprom almacena el resultado del promedio de sueldos y tampoco debe ser previamente inicializada, pues no tiene en cuenta ningún valor anterior.

Por último tenemos la impresión de resultados.

Ejercicio 2

Una empresa conoce para cada empleado los siguientes datos:

-nombre.
-sueldo.
-categoría.

Hay 100 empleados distribuidos en tres categorías.
Se pide calcular e imprimir:

a) Total de sueldos en pesos, que paga la empresa.
b) Cantidad de empleados que ganan más de 2000$.

c) Cantidad de empleados que ganan menos de 500$ cuya categoría sea 1.
d) Nombre del empleado que gana más.
e) Sueldo máximo.
f) Total de sueldos en pesos de cada categoría.
g) Sueldo promedio.

```
COMIENZO
totsdo = 0: mas2000 = 0: menos500 = 0
cat1 = 0 cat2 = 0: cat3 = 0
smax = 0
PARA empleado = 1 A 100
        INGRESAR "Nombre:", nom$
        INGRESAR "Sueldo:", sueldo
        INGRESAR "Categoría:", cat
        SELECCIONAR CASO cat
                CASO = 1
                    cat1 = cat1 + sueldo
                    SI sueldo < 500 ENTONCES menos500 = menos500 + 1
                CASO = 2
                    cat2 = cat 2+ sueldo
                CASO = 3
                    cat3 = cat3 + sueldo
        FIN SELECCIONAR
        totsdo = totsdo + sueldo
        SI sueldo > 2000 ENTONCES mas2000 = mas2000 + 1
        SI sueldo > smax ENTONCES
            smax = sueldo
            nomax$ = nom$
        FIN SI
PROXIMO
sprom = totsdo / 100
IMPRIMIR "Total de sueldos que paga la empresa:", totsdo, "$"
IMPRIMIR "Empleados que ganan más de 2000$:", mas2000
IMPRIMIR "Empleados de categoría1 que ganan menos de 500$:",
menos500
IMPRIMIR "Empleado que gana más:", nomax$
IMPRIMIR "Sueldo máximo:", smax, "$"
IMPRIMIR "Total de sueldos de la categoría 1:", cat1, "$"
```



```
IMPRIMIR "Total de sueldos de la categoría 2:", cat2, "$"
IMPRIMIR "Total de sueldos de la categoría 3:", cat3, "$"
IMPRIMIR "Sueldo promedio:", sprom, "$"
FIN
```

Se utilizaron dos contadores: mas2000, menos500; y cuatro acumuladores: totsdo, cat1, cat2 y cat3. Además se trabajó con smax para asignar el máximo.

De acuerdo al enunciado del ejercicio, el ingreso de datos se realizó dentro de un ciclo PARA de 100 repeticiones, una para cada empleado.

Con la decisión múltiple SELECCIONAR CASO, se evaluó la variable cat para acumular los sueldos de acuerdo a la categoría. En el caso particular de cat = 1, se incrementó el contador menos500 en los casos que el sueldo era inferior a 500$.

Con totsdo se acumularon todos los sueldos independientemente de la categoría. Este acumulador se podría haber evitado asignando directamente la sumas parciales de sueldo por categoría, almacenadas en cada uno de los acumuladores cat1, cat2 y cat3.

El promedio se obtiene de dividir el total acumulado por la cantidad de empleados, asignando el resultado en la variable sprom.

Ejercicio 3

Una línea aérea vende pasajes en 3 aeropuertos. En cada uno de ellos hay tres empleados que son los que efectúan las ventas. Cada vez que un cliente compra pasajes, se registran los siguientes datos:

- Número de aeropuerto.
- Número de empleado.
- Valor del pasaje.
- Cantidad de pasajes.

El ingreso de datos finaliza con un número de aeropuerto igual a cero.
Los números de los empleados se identifican del 1 al 9. Cada cliente puede comprar más de un pasaje.
Se pide calcular e imprimir:

a) La cantidad de pasajes vendidos por cada empleado.
b) La recaudación por aeropuerto.

c) El número de empleado que haya vendido mayor cantidad de pasajes en una venta.
d) La cantidad de pasajes vendidos por aeropuerto.
e) El porcentaje de ventas en pesos de cada empleado, sobre el total.
f) La cantidad de ventas que hayan excedido los 5000$

```
COMIENZO
cpe1 = 0:cpe2 = 0:cpe3 = 0:cpe4 = 0:cpe5 = 0:cpe6 = 0:cpe7 = 0:cpe8 = 0:cpe9 = 0
ra1 = 0: ra2 = 0: ra3 = 0
re1 = 0: re2 = 0: re3 = 0: re4 = 0: re5 = 0: re6 = 0: re7 = 0: re8 = 0: re9 = 0
pa1 = 0: pa2 = 0: pa3 = 0
c500 = 0: cpmax = 0
        INGRESAR "Número de aeropuerto:", na
        HACER MIENTRAS na <> 0
        INGRESAR "Número de empleado:", ne
        INGRESAR "Valor del pasaje:", vp
        INGRESAR "Cantidad de pasajes:", cp
        vta = cp * vp
        SELECCIONAR CASO ne
                CASO = 1
                        cpe1 = cpe1 + cp
                        re1 = re1 + vta
                CASO = 2
                        cpe2 = cpe2 + cp
                        re2 = re2 + vta
                CASO = 3
                        cpe3 = cpe3 + cp
                        re3 = re3 + vta
                CASO = 4
                        cpe4 = cpe4 + cp
                        re4 = re4 + vta
                CASO = 5
                        cpe5 = cpe5 + cp
                        re5 = re5 + vta
                CASO = 6
                        cpe6 = cpe6 + cp
                        re6 = re6 + vta
                CASO = 7
                        cpe7 = cpe7 + cp
```


Ejercicios de Propósito General

```
                        re7 = re7 + vta
                CASO = 8
                        cpe8 = cpe8 + cp
                        re8 = re8 + vta
                CASO = 9
                        cpe9 = cpe9 + cp
                        re9 = re9 + vta
        FIN SELECCIONAR
        SELECCIONAR CASO ra
                CASO = 1
                        ra1 = ra1 + vta
                        pa1 = pc1 + cp
                CASO = 2
                        ra2 = ra2 + vta
                        pa2 = pc2 + cp
                CASO = 3
                        ra3 = ra3 + vta
                        pa3 = pc3 + cp
        FIN SELECCIONAR
        SI cp > cpmax ENTONCES
                cpmax = cp
                nemax = ne
        FIN SI
        SI vta > 5000 ENTONCES c5000 = c5000 + 1
        INGRESAR " Número de aeropuerto:", na
REPETIR
rt = ra1 + ra2 + ra3
por1 = re1 / rt * 100
por2 = re2 / rt * 100
por3 = re3 / rt * 100
por4 = re4 / rt * 100
por5 = re5 / rt * 100
por6 = re6 / rt * 100
por7 = re7 / rt * 100
por8 = re8 / rt * 100
por9 = re9 / rt * 100
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 1:", cpe1
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 2:", cpe2
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 3:", cpe3
```

```
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 4:", cpe4
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 5:", cpe5
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 6:", cpe6
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 7:", cpe7
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 8:", cpe8
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 9:", cpe9
IMPRIMIR "Recaudación del aeropuerto 1:", ra1, "$"
IMPRIMIR "Recaudación del aeropuerto 2:", ra2, "$"
IMPRIMIR "Recaudación del aeropuerto 3:", ra3, "$"
IMPRIMIR "El empleado ", nemax, "fue el que más pasajes vendió."
IMPRIMIR "Cantidad de pasajes vendidos en el aeropuerto 1:", pa1
IMPRIMIR "Cantidad de pasajes vendidos en el aeropuerto 2:", pa2
IMPRIMIR "Cantidad de pasajes vendidos en el aeropuerto 3:", pa3
IMPRIMIR "Porcentaje de ventas del empleado 1:", por1, "%"
IMPRIMIR "Porcentaje de ventas del empleado 2:", por2, "%"
IMPRIMIR "Porcentaje de ventas del empleado 3:", por3, "%"
IMPRIMIR "Porcentaje de ventas del empleado 4:", por4, "%"
IMPRIMIR "Porcentaje de ventas del empleado 5:", por5, "%"
IMPRIMIR "Porcentaje de ventas del empleado 6:", por6, "%"
IMPRIMIR "Porcentaje de ventas del empleado 7:", por7, "%"
IMPRIMIR "Porcentaje de ventas del empleado 8:", por8, "%"
IMPRIMIR "Porcentaje de ventas del empleado 9:", por9, "%"
IMPRIMIR "Ventas mayores a 5000$:", c5000
FIN
```

En este ejercicio se trabajó con 24 acumuladores para acumular la cantidad de pasajes y la recaudación por empleado (cpe y re) y por aeropuerto (pa y ra). Además se utilizó sólo un contador (c5000).

A diferencia del ejercicio anterior, en este se utilizó un ciclo HACER MIENTRAS ya que se desconocía la cantidad de ventas a procesar.

El corte del ciclo está controlado por la variable na correspondiente a el número del aeropuerto.

Podemos apreciar que se repite dos veces la instrucción de ingreso del valor de na: primero para arrancar el ciclo se la coloca fuera de él. Luego al finalizar el ciclo, antes de REPETIR, para volver a evaluar la condición de corte.

Se utilizaron dos selecciones múltiples, una por empleado y otra por aeropuerto.

La recaudación total rt, se obtuvo como suma de las recaudaciones por

aeropuerto, pero también se podía haber calculado como suma de las recaudaciones de cada empleado, o bien utilizando otro acumulador que sume venta a venta realizada en cada repetición del ciclo.

Hay un cálculo de máximo (cpmax) a partir del cual, se imprime el número de empleado relacionado con este.

El único contador es utilizado para contar la cantidad de ventas mayores a 5000$.

Los porcentajes se calculan como el cociente entre el valor individual sobre el valor total multiplicado todo por 100. Al igual que las variables utilizadas para almacenar promedios, las variables a las cuales se les asigna el cálculo de los porcentajes, no necesitan ser inicializadas al comienzo del programa pues no tienen en cuenta valores previos.

Ejercicio 4

Una empresa de aliscafos, vende pasajes en tres terminales. En cada una de ellas hay dos empleados que son los que efectúan las ventas. Cada vez que una persona compra pasajes, se registran los siguientes datos:

- Número de terminal.
- Número de empleado.
- Valor del pasaje.
- Cantidad de pasajes.

Se deberán procesar 1500 ventas. Los empleados se numeran del 1 al 6. Cada cliente puede comprar más de un pasaje.

Se pide calcular e imprimir:

a) La cantidad de pasajes vendidos por cada empleado.
b) La recaudación por terminal.
c) El número de empleado que haya vendido menor cantidad de pasajes a un cliente.
d) La cantidad de pasajes vendidos por terminal.
e) El porcentaje de ventas en cantidad de pasajes, de cada terminal sobre el total.
f) La cantidad de ventas por debajo de los 200$.

```
COMIENZO
cpe1 = 0:cpe2 = 0:cpe3 = 0:cpe4 = 0:cpe5 = 0:cpe6 = 0
rt1 = 0: rt2 = 0: rt3 = 0
pt1 = 0: pt2 = 0: pt3 = 0
menos200 = 0: cpmin = 9999999999
PARA ventas = 1 A 1500
        INGRESAR "Número de terminal:", nt
        INGRESAR "Número de empleado:", ne
        INGRESAR "Valor del pasaje:", vp
        INGRESAR "Cantidad de pasajes:", cp
        vta = cp * vp
        SELECCIONAR CASO ne
                CASO = 1
                        cpe1 = cpe1 + cp
                CASO = 2
                        cpe2 = cpe2 + cp
                CASO = 3
                        cpe3 = cpe3 + cp
                CASO = 4
                        cpe4 = cpe4 + cp
                CASO = 5
                        cpe5 = cpe5 + cp
                CASO = 6
                        cpe6 = cpe6 + cp
        FIN SELECCIONAR
        SELECCIONAR CASO nt
                CASO = 1
                        rt1 = rt1 + vta
                        pt1 = pt1 + cp
                CASO = 2
                        rt2 = rt2 + vta
                        pt2 = pt2 + cp
                CASO = 3
                        rt3 = rt3 + vta
                        pt3 = pt3 + cp
        FIN SELECCIONAR
        SI cp < cpmin ENTONCES
                cpmin = cp
                nemin = ne
        FIN SI
```



```
        SI vta < 200 ENTONCES menos200 = menos200 + 1
PROXIMO
pt = pt1 + pt2 + pt3
por1 = pt1 / pt * 100
por2 = pt2 / pt * 100
por3 = pt3 / pt * 100
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 1:", cpe1
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 2:", cpe2
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 3:", cpe3
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 4:", cpe4
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 5:", cpe5
IMPRIMIR "Cantidad de pasajes vendidos por el empleado 6:", cpe6
IMPRIMIR "Recaudación de la terminal 1:", rt1, "$"
IMPRIMIR "Recaudación de la terminal 2:", rt2, "$"
IMPRIMIR "Recaudación de la terminal 3:", rt3, "$"
IMPRIMIR "El empleado ", nemin, "fue el que menos pasajes vendió."
IMPRIMIR "Cantidad de pasajes vendidos en la terminal 1:", pt1
IMPRIMIR "Cantidad de pasajes vendidos en la terminal 2:", pt2
IMPRIMIR "Cantidad de pasajes vendidos en la terminal 3:", pt3
IMPRIMIR "Porcentaje de ventas de la terminal 1:", por1, "%"
IMPRIMIR "Porcentaje de ventas de la terminal 2:", por2, "%"
IMPRIMIR "Porcentaje de ventas de la terminal 3:", por3, "%"
IMPRIMIR "Ventas menores a 200$:", menos200
FIN
```

Se utilizaron en este ejercicio 12 acumuladores: 6 para los pasajes vendidos por cada empleado (cpe) y 6 para los pasajes (pt) y la recaudación (rt) por terminal.

la variable menos200 es el único contador que permite determinar la cantidad de ventas menores a doscientos pesos.

la variable cpmin almacena el la mínima venta registrada en cantidad de pasajes por un empleado y funciona en forma paralela a la variable nemin, que almacenará el número de empleado que realizó dicha venta. **Se utilizó aleatoriamente el valor 9999999999 para inicializar cpmin, esta puede ser iniciada con cualquier valor lo suficientemente grande como para asegurar que en una primera pasada por el ciclo, el primer valor leído irá a parar a cpmin.**

Se trabajó nuevamente con dos decisiones o selecciones múltiples: una para los empleados y otra para las terminales.

Los tres porcentajes pedidos, se calcularon por cantidad de pasajes vendidos.

Ejercicio 5

Una casa de comidas tiene 6 mesas y 3 mozos. Cada mozo atiende dos mesas y cuando la mesa se desocupa se registran los siguientes datos:

- Número de mozo.
- Número de mesa.
- Importe de la cuenta.
- Cantidad de personas que comieron.

La cantidad de cuentas a procesar es de 500.
Se pide calcular e imprimir:

a) La cantidad de personas atendidas por cada mozo.
b) La recaudación por mozo.
c) El número de mozo que haya tenido la cuenta de menor importe.
d) La cantidad de personas que comieron por mesa.
e) El porcentaje de ventas en pesos, de cada mozo sobre el total.
f) La cantidad de cuentas con importe por encima de los 100$.

```
COMIENZO
cpamo1 = 0: cpamo2 = 0: cpamo3 = 0
recmo1 = 0: recmo2 = 0: recmo3 = 0
cuentmin = 9999999999
cpme1 = 0: cpme2 = 0: cpme3 = 0: cpme4 = 0: cpme5 = 0: cpme6 = 0
mas100 = 0
PARA ventas = 1 A 500
        INGRESAR "Número de mozo:", numozo
        INGRESAR "Número de mesa:", numesa
        INGRESAR "Importe de la cuenta:", imp
        INGRESAR "Cantidad de personas que comieron:", cp
        SELECCIONAR CASO numozo
                CASO = 1
                        cpamo1 = cpamo1 + cp
                        recmo1 = recmo1 + imp
                CASO = 2
```



```
                        cpamo2 = cpamo2 + cp
                        recmo2 = recmo2 + imp
                CASO = 3
                        cpamo3 = cpamo3 + cp
                        recmo3 = recmo3 + imp
        FIN SELECCIONAR
        SELECCIONAR CASO numesa
                CASO = 1
                        cpme1 = cpme1 + cp
                CASO = 2
                        cpme2 = cpme2 + cp
                CASO = 3
                        cpme3 = cpme3 + cp
                CASO = 4
                        cpme4 = cpme4 + cp
                CASO = 5
                        cpme5 = cpme5 + cp
                CASO = 6
                        cpme6 = cpme6 + cp
        FIN SELECCIONAR
        SI imp < cuentmin ENTONCES
                cuentmin = imp
                mozomin = numozo
        FIN SI
        SI imp > 100 ENTONCES mas200 = mas200 + 1
PROXIMO
total = recmo1 + recmo2 + recmo3
por1 = recmo1 / total * 100
por2 = recmo2 / total * 100
por3 = recmo3 / total * 100
IMPRIMIR "Personas atendidas por el mozo 1:", cpamo1
IMPRIMIR "Personas atendidas por el mozo 2:", cpamo2
IMPRIMIR "Personas atendidas por el mozo 3:", cpamo3
IMPRIMIR "Recaudación del mozo 1:", recmo1,"$"
IMPRIMIR "Recaudación del mozo 2:", recmo2,"$"
IMPRIMIR "Recaudación del mozo 3:", recmo3,"$"
IMPRIMIR "Número de mozo que tuvo la menor cuenta:", mozomin
IMPRIMIR "Cantidad de personas atendidas en la mesa 1:", cpme1
IMPRIMIR "Cantidad de personas atendidas en la mesa 2:", cpme2
```

```
IMPRIMIR "Cantidad de personas atendidas en la mesa 3:", cpme3
IMPRIMIR "Cantidad de personas atendidas en la mesa 4:", cpme4
IMPRIMIR "Cantidad de personas atendidas en la mesa 5:", cpme5
IMPRIMIR "Cantidad de personas atendidas en la mesa 6:", cpme6
IMPRIMIR "Porcentaje de las ventas del mozo 1:", por1,"%"
IMPRIMIR "Porcentaje de las ventas del mozo 2:", por2,"%"
IMPRIMIR "Porcentaje de las ventas del mozo 3:", por3,"%"
IMPRIMIR "Cantidad de cuentas con importe mayor a 100$:", mas100
FIN
```

Tenemos en ese caso 12 acumuladores, 6 para la cantidad de personas atendidas por mesa (cpme) y 6 para la recaudación (recmo) y la cantidad de personas atendidas (cpamo) por mozo.

Se trabajó nuevamente con sólo un contador, para la cantidad de cuentas con importe por encima de 100$.

La variable cuentmin almacena el importe mínimo para una cuenta y trabaja asociada a mozomin que guarda el número de mozo responsable de la atención de dicha cuenta.

Los porcentajes están calculados en función de las recaudaciones por mozo.

RESUMEN

Recordaremos algunas consideraciones prácticas surgidas del análisis de los ejercicios:

✓ Lo primero que debemos hacer cuando comenzamos a escribir un programa, es inicializar todas las variables que vamos a utilizar como: contadores, acumuladores, máximos y mínimos.

✓ Por una cuestión práctica con respecto a la escritura, es conveniente separar estas asignaciones iniciales por dos puntos (:), para no tener que cambiar permanentemente de línea de instrucción. Esto último es válido en la mayoría de los lenguajes de programación.

✓ Las variables que trabajan asociadas a máximos o mínimos, no es necesario inicializarlas en el comienzo, pues en realidad no almacenan valores máximos o mínimos, sino la variable a la cual pertenece dicho valor.

✓ La variables que almacenan el resultado de promedios, tampoco deben ser previamente inicializadas, pues no tienen en cuenta ningún valor anterior.

✓ El promedio se obtiene de dividir el total acumulado por la cantidad de valores considerados.

✓ En los ciclos HACER MIENTRAS se repite dos veces la instrucción de ingreso del valor de la variable que condiciona su acceso: primero para arrancar el ciclo se la coloca fuera de él. Luego al finalizar el ciclo, antes de REPETIR, para volver a evaluar la condición de corte.

✓ Los porcentajes se calculan como el cociente entre el valor individual sobre el valor total multiplicado todo por 100.

✓ Al igual que las variables utilizadas para almacenar promedios, las

variables a las cuales se les asigna el cálculo de los porcentajes, no necesitan ser inicializadas al comienzo del programa pues no tienen en cuenta valores previos.

✓ Se utilizó aleatoriamente el valor 9999999999 para inicializar una variable mínimo, esta puede ser iniciada con cualquier valor lo suficientemente grande como para asegurar que en una primera pasada por el ciclo, el primer valor leído irá a parar a la variable que almacena el mínimo



Ejercicios Propuestos



1. Una empresa textil desea procesar sus ventas. Cada vez que una persona realiza una compra se le entrega una factura donde consta:

- Número de factura.
- Código del artículo.
- Cantidad del artículo.
- Precio unitario.

En cada factura se registra un solo código de artículo. Los artículos son cuatro. El ingreso de datos finaliza con un número de factura igual a cero.
Se pide calcular e imprimir:

a) Total general facturado en pesos.
b) Cantidad de unidades vendida para cada uno de los artículos.
c) Total de artículos vendidos.
d) Cantidad de facturas emitidas para cada uno de los artículos.
e) Número de factura con mayor importe en pesos.
f) Número de artículo con menor cantidad pedida en una factura.
g) Porcentaje de ventas en pesos de cada uno de los artículos sobre el total.

2. Un supermercado tiene tres secciones:

- Almacén.
- Verdulería.
- Panadería.

El supermercado tiene en total seis cajas, dos para cada sección. Cada vez que una persona pasa por la caja, se le entrega un comprobante con los siguientes datos :

- Número de caja.
- Número de sección.
- Importe de la venta.
- Cantidad de productos comprados.

Si el importe de la venta supera los 50$, se le hace un descuento del 10% sobre el total.

Un comprobante similar al que se le entrega a los clientes queda registrado en la cinta de la máquina registradora.

Con dichos datos se desea saber:

a) La cantidad de comprobantes emitidos en cada caja.
b) La recaudación de cada caja.
c) La caja en la que se registró la venta de mayor importe.
d) El porcentaje de ventas en pesos de cada caja, sobre el total de las ventas.
e) La cantidad de comprobantes con importe mayor a 100$, que hayan comprado menos de tres productos.
f) El sueldo de cada cajero (cada cajero gana el 10% de la recaudación de su caja).

3. Una cartelera de teatros vende entradas para distintas obras, en cada una de sus dos sucursales.

En la actualidad, las obras que se ofrecen son cuatro:

1 Mi bella dama........... .25$
2 El amateur.................15$
3 101 dálmatas............ .20$
4 Los miserables...........30$

Para que el procesamiento de la información sea claro, si una persona compra entradas para dos obras distintas, se le emiten dos comprobantes distintos. Si compra varias entradas para la misma obra, se le entrega un sólo comprobante, donde se indica cuantas compra.

En cada comprobante se registra:

- Número de sucursal.
- Número de obra.
- Cantidad de lugares.

Si compra más de 10 entradas para una misma obra, se le aplica un descuento del 15% sobre el precio total.

Se pide calcular e imprimir:

c) La cantidad de entradas vendidas para cada obra.
b) La recaudación por sucursal.
c) El número de sucursal que haya vendido mayor cantidad de entradas en una venta.
c) La ganancia de la cartelera (le queda el 20% de su recaudación).
e) El porcentaje de ventas, en cantidad de entradas, de cada sucursal.
f) La cantidad de entradas vendidas en la sucursal 1 de "Mi bella dama".

4. Un noticiero de televisión tiene tres comentaristas políticos que cubren alternadamente:

1- Casa de gobierno.
2- Ministerio de Economía.
3- Ministerio de Cultura y Educación.
4- Ministerio de Trabajo.

Los comentaristas reciben su paga por horas, siendo el valor de estas:

Comentarista 1....15$
Comentarista 2....20$
Comentarista 3....25$

Al terminar una jornada de trabajo cada uno de los comentaristas, entrega en el canal una planilla con los siguientes datos:

- Número de comentarista.
- Número del lugar del trabajo.
- Cantidad de horas trabajadas.
- Cantidad de personas entrevistadas.

Si la cantidad de horas trabajadas en un día es mayor que 10, se le entrega un plus de 40$.

El canal cuenta a fin de mes con todas las planillas.

Se pide calcular e imprimir:

a) La cantidad de personas entrevistadas por cada comentarista.
b) El sueldo de cada comentarista.
c) La cantidad de horas trabajadas en cada lugar de trabajo.
d) El número de comentarista con mayor cantidad de horas trabajadas, en un día de trabajo.

c) El sueldo promedio.
f) La cantidad de veces que los comentaristas trabajaron más de 15 horas en casa de gobierno, en un día de trabajo.

5. Una empresa de micros vende pasajes a tres destinos del interior del país:

1- Córdoba.
2- Mendoza.
3- Tucumán.

Los micros tienen dos clases y por consiguiente dos tarifas diferentes:

1- Turista........50$.
2- Primera. ....70$

El costo del pasaje es el mismo para los tres destinos.
Al pasajero se le entrega un ticket, donde se consignan los siguientes datos:

- Número de ticket.
- Número de destino.
- Número de clase.
- Cantidad de pasajes.

Si una persona compra más de 15 pasajes, se le descuenta un 20% del precio total.
Se pide calcular e imprimir:

a) La cantidad de pasajes vendidos a cada destino.
b) La recaudación por clase de pasajes.
c) El número de ticket con mayor importe, en una venta.
d) La cantidad de pasajes vendidos por clase.
e) El porcentaje de ventas, en cantidad de pasajes, de cada destino.
f) La cantidad de pasajes vendidos a Córdoba en primera clase .

6. Un restaurante tiene 6 mesas y 3 mozos. Cada mozo atiende dos mesas y cuando la mesa se desocupa se registran los siguientes datos:

- Número de mozo.
- Número de mesa.
- Importe de la cuenta.
- Cantidad de personas que comieron.

El ingreso de datos finaliza con un número de mozo igual a cero.
Se pide calcular e imprimir:

a) La cantidad de personas atendidas por cada mozo.
b) La recaudación por mozo.
c) El número de mozo que haya tenido la cuenta de mayor importe.
d) La cantidad de personas que comieron por mesa.
e) El porcentaje de ventas en pesos, de cada mozo sobre el total.
f) La cantidad de cuentas con importe por debajo de los 50$.



Vectores

5.1 INTRODUCCIÓN

Hasta el momento, para hacer mención a un dato utilizábamos una variable.

La dificultad se presenta cuando tenemos una gran cantidad de datos que están relacionados entre sí. Para cada uno de esos datos se debería definir una variable distinta, lo que ocasiona gran dificultad cuando tenemos que escribir un programa, debido a la cantidad de variables a utilizar.

Tomemos como ejemplo el ejercicio 3 desarrollado en el capítulo 4:

"Una línea aérea vende pasajes en 3 aeropuertos. En cada uno de ellos hay tres empleados que son los que efectúan las ventas."

Cuando calculamos las ventas por empleado utilizamos 9 acumuladores, desde cpe1 hasta cpe9, pues teníamos 3empleados por cada uno de los 3 aeropuertos.

¿Qué sucedería si en lugar de tener 3 aeropuertos nos hubiéramos referido a 30?

Necesitaríamos 90 acumuladores, lo que acarrearía dificultades para escribir las 90 variables cpe.

Para resolver estas dificultades se agrupan los datos en un mismo conjunto, bajo un nombre común, que se pueden tratar como una sola unidad.

A estos conjuntos se los denomina estructuras de datos.

Las estructuras de datos, de acuerdo al lugar donde se almacenan, se clasifican en:

✓ **Internas**: se almacenan en la memoria de la computadora, se clasifican en vectores y matrices según el tipo.

✓ **Externas**: se almacenan en soportes externos como los discos; se denominan archivos.

En este capítulo estudiaremos los vectores.

5.2 CONCEPTO DE VECTOR

Un vector es un conjunto de datos del mismo tipo, numéricos o alfanuméricos, organizados bajo un mismo nombre y almacenados en la memoria de la computadora.

Cada dato se denomina elemento del vector y está referenciado por una variable **indice**, que indica la posición de dicho elemento dentro del vector.

Previo al uso de un vector, hay que reservar una zona de memoria para su uso, así como definir el número de elementos necesarios para acceder a cada elemento de la estructura. Al proceso de reservar una zona de la memoria para almacenar los datos del vector, se lo denomina dimensionar y se realiza de la siguiente forma:

Sintaxis:	***Nombre del vector(cantidad de elementos)***
Ejemplo:	***contador(20)***

El ejemplo indica que el vector de nombre contador, está compuesto por 20 elementos y por consiguiente puede almacenar 20 datos bajo un mismo nombre.

La cantidad de elementos es el máximo valor que puede tomar el índice.

Los vectores se dimensionan una sola vez al comienzo del programa, para reservar la zona de memoria que se va a utilizar.

Los índices sólo pueden tomar valores enteros. Si el índice sobrepasa la dimensión del vector se producirá un error.

La posición del elemento al que queremos acceder, va entre paréntesis. Debemos diferenciar perfectamente los conceptos de posición y elemento. Analicemos el ejemplo de la figura1:

Posición (índice)	num (5)
1	25
2	32
3	78
4	84
5	93

Figura 1

En la figura uno tenemos representado el vector num compuesto por 5 elementos:

num(1) = 25
num(2) = 32
num(3) = 78
num(4) = 84
num(5) = 93

- La **posición** es la dirección del lugar que ocupa cada dato dentro del vector y está indicado por el índice: en el ejemplo serían 1, 2, 3, 4 y 5.
- El **elemento** es el dato que hay almacenado en el vector, en esa posición: en nuestro ejemplo los datos 25, 32, 78, 84, 93.

Los vectores, al residir en la memoria de la computadora, son de acceso directo, por lo que podemos situarnos directamente en la posición deseada. Los elementos de un vector se utilizan en forma análoga a la de cualquier variable; intervendrán en instrucciones de:

- ✗ asignación: variable = vector() o vector1(i) = vector(j)
- ✗ entrada: INGRESAR vector(i)
- ✗ salida: IMPRIMIR vector(i)
- ✗ contador: vector(i) = vector(i) + constante
- ✗ acumulador: vector(i) = vector(i) + variable

Los vectores que contienen datos alfanuméricos, deben escribirse con el símbolo $ al final: vector$(i).

5.3 INICIALIZACIÓN DE UN VECTOR

En los vectores siempre conocemos de antemano los límites entre los que varía el índice para realizar su recorrido. Esto es posible porque lo primero que debemos hacer para crear un vector es dimensionarlo. Así si queremos colocar en cero el vector de nombre contador de 20 elementos lo haríamos de la siguiente manera:

```
contador(20)
PARA i = 1 A 20
        contador(i) = 0
PROXIMO
```

Como podemos apreciar se recurre a un ciclo PARA.... PROXIMO, para colocar en cero cada uno de los elementos del vector, de esta manera se repetirá 20 veces la asignación.

5.4 CARGA DE UN VECTOR

De la misma manera analizada en el punto anterior, recurrimos a un ciclo PARA....PROXIMO, para cargar un vector durante el ingreso de datos.

Consideremos ahora el vector nombre$ de 10 elementos en el cual vamos a cargar 10 nombres que ingresaremos desde afuera del programa:

```
nombre$(10)
PARA i = 1 A 10
        INGRESAR "Ingrese el nombre:", nombre$(i)
PROXIMO
```

En este caso, se repetirá 10 veces el ingreso de nombres, que se irán cargando en un orden secuencial, en cada uno de las posiciones del vector.

5.5 EJERCICIOS DE APLICACIÓN

Desarrollaremos ahora un conjunto de ejercicios que nos posibilitarár un mejor entendimiento del funcionamiento de un vector y de su gran utilidad como herramienta de programación.

Ejercicio 1

Cargar un vector de 50 elementos e imprimir:

a) El cuarto elemento.
b) El segundo elemento.
c) Los elementos en orden invertido.
d) El producto entre el primer elemento y el último.
e) Los elementos de índice par.
f) Los elementos de índice impar.

```
COMIENZO
vec(50)
PARA i = 1 A 50
        INGRESAR "Ingrese un número:", vec(i)
PROXIMO
IMPRIMIR "Cuarto elemento:", vec(4)
IMPRIMIR "Segundo elemento:", vec(2)
PARA i = 50 A 1 PASO -1
        IMPRIMIR vec(i)
PROXIMO
IMPRIMIR "Producto:", vec(1) * vec(50)
PARA i = 2 A 50 PASO 2
        IMPRIMIR vec(i)
PROXIMO
PARA i = 1 A 49 PASO 2
        IMPRIMIR vec(i)
PROXIMO
FIN
```

Se puede apreciar como es válido imprimir un elemento particular de un vector:

```
IMPRIMIR "Cuarto elemento:", vec(4)
IMPRIMIR "Segundo elemento:", vec(2)
```

También es correcto imprimir un vector en forma invertida o alternadamente para lo cual habrá que tener especial cuidado con la definición del PASO en el ciclo PARA.

Ejercicio 2

Cargar un vector de n elementos con números enteros e imprimir:

c) Cantidad de números positivos.
b) Cantidad de números negativos.
c) Cantidad de números menores de 25.

```
COMIENZO
INGRESAR "Ingrese la cantidad de elementos:", n
vec(n) : pos = 0 : neg = 0 : menor25 = 0
PARA i = 1 A n
        INGRESAR "Ingrese un número:", vec(i)
        SI vec(i) > 0 ENTONCES pos = pos + 1
        SI vec(i) < 0 ENTONCES neg = neg + 1
        SI vec(i) < 25 ENTONCES menor25 = menor25 + 1
PROXIMO
IMPRIMIR "Cantidad de números positivos:", pos
IMPRIMIR "Cantidad de números negativos:", neg
IMPRIMIR "Cantidad de números menores a 25:", menor25
FIN
```

Este ejercicio presenta como variante que la cantidad de elementos del vector debe ser ingresada previo al dimensionamiento del mismo:

```
INGRESAR "Ingrese la cantidad de elementos:", n
vec(n)
```

Ejercicio 3

Dadas 25 edades, cargarlas en un vector y calcular e imprimir:

a) Edad promedio.

b) Cantidad de edades mayores a 18 años.

c) Todas las edades mayores al promedio.


```
COMIENZO
sum = 0 : mayor18 = 0
edad(25)
PARA i = 1 A 25
        INGRESAR "Ingresar edad:", edad(i)
        sum = sum + edad(i)
        SI edad(i) > 18 ENTONCES mayor18 = mayor18 + 1
PROXIMO
edprom = sum / 25
IMPRIMIR "Edad promedio:", edprom,"años"
IMPRIMIR "Cantidad de edades mayores a 18 años:", mayor18
IMPRIMIR "Edades mayores al promedio:"
PARA i = 1 A 25
        SI edad(i) > edprom ENTONCES IMPRIMIR edad(i)
PROXIMO
FIN
```

En este ejemplo se utiliza el vector como variable a sumar en el acumulador sum:

sum = sum + edad(i)

También dada la cantidad de procesos, se podría haber realizado cada cálculo en forma independiente:

```
COMIENZO
sum = 0 : mayor18 = 0
edad(25)

PARA i = 1 A 25
        INGRESAR "Ingresar edad:", edad(i)
PROXIMO

PARA i = 1 A 25
        sum = sum + edad(i)
PROXIMO
edprom = sum / 25
PARA i = 1 A 25
        SI edad(i) > 18 ENTONCES mayor18 = mayor18 + 1
PROXIMO
```

```
IMPRIMIR "Edad promedio:", edprom,"años"
IMPRIMIR "Cantidad de edades mayores a 18 años:", mayor18
IMPRIMIR "Edades mayores al promedio:"
PARA i = 1 A 25
        SI edad(i) > edprom ENTONCES IMPRIMIR edad(i)
PROXIMO
FIN
```

La segunda forma de resolver el ejercicio será tema de estudio en el próximo capítulo

Ejercicio 4

Una empresa desea procesar sueldos de sus empleados. Para cada empleado se conoce sueldo y nombre.
Se desea saber:

a) Nombre del o los empleados con sueldo mayor al sueldo promedio.
b) Cantidad de empleados que ganan más de 500$.

```
COMIENZO
INGRESAR " Ingrese la cantidad de empleados:", n
sueldo(n) : nom$(n)
totsdo = 0 : mas500 = 0
PARA i = 1 A n
        INGRESAR "Ingrese el nombre:", nom$(i)
        INGRESAR "Ingrese el sueldo:", sueldo(i)
PROXIMO
PARA i = 1 A n
        totsdo = totsdo + sueldo(i)
        SI sueldo(i) > 500 ENTONCES mas500 = mas500 + 1
PROXIMO
sdoprom = totsdo / n
IMPRIMIR "Empleados con sueldo mayor al sueldo promedio:"
PARA i = 1 A n
        SI sueldo(i) > sdoprom ENTONCES IMPRIMIR nom$(i)
PROXIMO
IMPRIMIR "Cantidad de empleados que ganan más de 500$:", mas500
FIN
```


Este ejercicio nos presenta como variante, la impresión del vector nom$, condicionada por el vector sueldo(i):

SI sueldo(i) > sdoprom ENTONCES IMPRIMIR nom$(i)

5.6 MÁXIMOS Y MÍNIMOS DE UN VECTOR

Anteriormente nos referimos a la búsqueda de máximos y mínimos entre un conjunto de datos.

En los ejercicios desarrollados a continuación veremos como se facilita dicha búsqueda dentro de un vector.

Ejercicio 5

Dado un vector de 30 elementos, imprimir el valor máximo (suponemos que el vector fue cargado previamente).

```
COMIENZO
vec(30)
vmax = vec(1)
PARA i = 2 A 30
        SI vec(i) > vmax ENTONCES vmax = vec(i)
PROXIMO
IMPRIMIR "Valor máximo:", vmax
FIN
```

La primera gran diferencia está en que no tenemos que colocar la variable vmax en cero, pues al disponer de todos los datos invocando al vector, se define inicialmente a vmax, con el dato almacenado en el primer elemento: vmax = vec(i).

A continuación se recorre el vector desde el segundo elemento hasta el último, utilizando un ciclo PARA. Esta búsqueda permite encontrar el valor máximo buscado en el ejercicio.

Ejercicio 6

Dado un vector de 10 elementos, imprimir el valor mínimo (suponemos que el vector fue cargado previamente).

```
COMIENZO
vec(10)
vmin = vec(1)
PARA i = 2 A n
        SI vec(i) < vmin ENTONCES vmin = vec(i)
PROXIMO
IMPRIMIR "Valor mínimo:", vmin
FIN
```

Tenemos una variante con respecto al ejercicio anterior: en este caso invertimos la pregunta dentro del ciclo para obtener el valor mínimo.

Ejercicio 7

Dados un vector sueldo y un vector nombre$, cargarlos con los sueldos y nombres de 30 empleados y luego imprimir:

a) Sueldo máximo.
b) Sueldo mínimo.
c) Nombre del empleado que gana más.
d) Nombre del empleado que gana menos.

```
COMIENZO
sueldo(30) : nombre$(30)
PARA i = 1 A 30
        INGRESAR "Ingrese el nombre:", nombre$(i)
        INGRESAR "Ingrese el sueldo:", sueldo(i)
PROXIMO
smax = sueldo(1)
smin = sueldo(1)
PARA i = 2 A 30
        SI sueldo(i) > smax ENTONCES smax = sueldo(i)
        SI sueldo(i) < smin ENTONCES smin = sueldo(i)
PROXIMO
IMPRIMIR "Sueldo máximo:", smax, "$"
IMPRIMIR "Sueldo mínimo:", smin, "$"
IMPRIMIR "Nombre de los empleados que ganan el sueldo máximo:"
PARA i = 1 A 30
        SI sueldo(i) = smax ENTONCES IMPRIMIR nombre$(i)
```



```
PROXIMO
IMPRIMIR "Nombre de los empleados que ganan el sueldo mínimo:"
PARA i = 1 A 30
        SI sueldo(i) = smin ENTONCES IMPRIMIR nombre$(i)
PROXIMO
FIN
```

En los procedimientos utilizados anteriormente, colocamos como restricción, que no podían existir máximos y mínimos múltiples; un sólo valor, entre un conjunto podía estar asociado al resultado de la búsqueda.

Una gran ventaja en la utilización de vectores es la posibilidad de poder encontrar más de una variable asociada a un máximo o a un mínimo, que responda a la búsqueda solicitada.

En el ejemplo, varios empleados pueden ganar el sueldo máximo o el mínimo; el vector nombre$, trabaja asociado con el de sueldos. La búsqueda de un sueldo máximo o mínimo, implica buscar también uno o más nombres que respondan a aquellos.

Si bien el valor máximo o mínimo sigue siendo uno sólo, al ser múltiple la posibilidad de encontrar varios elementos del vector nombre$ que tengan asociado el sueldo hallado como máximo o como mínimo, se denomina a estos valores hallados como máximos y mínimos múltiples.

La búsqueda se hace en dos etapas: primero se determina el sueldo máximo y el sueldo mínimo y luego se buscan todos los nombres asociados a estos valores.

5.7 ORDENAMIENTO DE VECTORES

Cuando se desean visualizar los elementos de una lista, en la mayoría de las ocasiones no interesa ver los datos como ingresaron, sino en forma ordenada, para lo cual es necesario ordenar el vector.

El ordenamiento puede ser:

✓ **Ascendente:** los elementos están situados de menor a mayor.
✓ **Descendente:** los elementos están situados de mayor a menor.

Los valores repetidos, en caso de existir, quedan en posiciones contiguas.

Para llevar a cabo el ordenamiento de un vector hay diversos métodos, que sirven tanto para las listas numéricas como para las alfanuméricas.

En el desarrollo de nuestra explicación utilizaremos el método del burbujeo

por considerar que a los fines de nuestra explicación tiene una buena relación entre lo didáctico, y la velocidad y eficiencia en su uso.

5.7.1 Método del Burbujeo

Se basa en llevar el máximo del vector a la última posición. Para lograr esto se van tomando los elementos de a dos y se los va comparando e intercambiando hasta cumplir con el objetivo. Además para optimizar su funcionamiento utilizaremos un switch que nos permitirá conocer si el vector está ordenado o no.

Dado un vector de 10 elementos, vamos a ordenarlo en forma ascendente:

```
COMIENZO
vec(10)
cota = 10
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                SI vec(i) > vec(i + 1) ENTONCES
                        aux = vec(i)
                        vec(i) = vec(i + 1)
                        vec( i + 1) = aux
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
FIN
```

Primeramente asignamos a la variable cota un valor igual a la cantidad de elementos del vector a ordenar; más adelante explicaremos su función.

La variable k es un switch que cumple con las siguientes condiciones:

Si k = 0, implica que el vector está ordenado; si k <> 0 el vector está desordenado.

Al comienzo del ejercicio suponemos que el vector está desordenado, lo

que indicamos colocando k en 1 (k podría haber sido igual a cualquier valor distinto de cero).

Esto hace que nos aseguremos la entrada en el ciclo HACER MIENTRAS al preguntar por k <> 0.

Una vez dentro del ciclo se coloca a k en 0. De esta manera estamos suponiendo que una vez que se ejecuten las instrucciones del ciclo HACER MIENTRAS, el vector quedará ordenado.

A continuación tenemos un ciclo PARA de valor inicial igual a uno y de valor final a cota-1 (PARA i = 1 A cota-1), lo cual coincide con la cantidad de elementos, menos uno, que tiene el vector.

Esto está relacionado con la primer instrucción dentro del ciclo PARA:

SI vec(i) > vec(i + 1) ENTONCES , que compara el primer elemento con el segundo, el segundo con el tercero, el tercero con el cuarto, etc, hasta llegar a comparar los dos últimos elementos. Si el ciclo fuera de 1 a cota, la última comparación la haría entre el elemento 10 (i = cota) y el inexistente elemento 11. Esto explica el motivo de recorrer hasta cota-1.

Cada vez que la condición es verdadera, hay que hacer un intercambio de valores:

```
aux = vec(i)
vec(i) = vec(i + 1)
vec( i + 1) = aux
```

Como se puede apreciar se utiliza una variable auxiliar aux, para no perder ningún dato al copiar de vec(i) a vec(i+1) y viceversa.

Luego aparece nuevamente una asignación a la variable k, esta vez del índice i:

```
k = i
```

El valor de i en el momento de la asignación corresponde a la posición de los últimos elementos intercambiados, lo que tiene un doble significado:

✓ Cargar k con un valor distinto de cero para volver a ejecutar el ciclo HACER MIENTRAS, pues el hecho de haber realizado un intercambio condiciona a que se ejecute el ciclo al menos una vez más para verificar si el vector quedó ordenado.

✓ Transferir fuera del ciclo PARA, el valor de k hacia cota (cota = k), para no tener que volver a comparar nuevamente desde el primer elemento hasta

el último, sino hasta el último intercambiado. Se supone que el resto del vector no tuvo intercambios porque está ordenado.

El vector estará ordenado cuando al ejecutar integramente el ciclo PARA, nunca se cumpla la condición verificada por la instrucción SI ENTONCES. Si esto sucede no se asignara i a k, permaneciendo esta última en cero, valor cargado en k luego de ingresar en el ciclo HACER MIENTRAS. De esta forma no se cumplirá la condición k <> 0 y finalizará el proceso de ordenamiento.

Supongamos que los elementos de la lista contengan los siguientes valores:

9 7 6 5 4 3 11 8 10 2

En una primera pasada por el ciclo PARA los elementos quedan de la siguiente forma:

7 6 5 4 3 9 8 10 2 11

En la siguiente pasada no hará falta comparar hasta el último elemento, porque en la anterior pasada ya hemos llevado el máximo a la última posición.

6 5 4 3 7 8 9 2 10 11

De igual forma procederemos con las siguientes pasadas:

5 4 3 6 7 8 2 9 10 11	tercera pasada.
4 3 5 6 7 2 8 9 10 11	cuarta pasada.
3 4 5 6 2 7 8 9 10 11	quinta pasada.
3 4 5 2 6 7 8 9 10 11	sexta pasada.
3 4 2 5 6 7 8 9 10 11	séptima pasada.
3 2 4 5 6 7 8 9 10 11	octava pasada.
2 3 4 5 6 7 8 9 10 11	novena pasada.
2 3 4 5 6 7 8 9 10 11	décima pasada.

La décima pasada arroja como resultado la misma ubicación de elementos que la novena lo cual indica que el vector está ordenado.

Ejercicio 8

Dado un vector de 24 elementos, ordenarlo en forma descendente

```
COMIENZO
vec(24)
cota = 24
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                SI vec(i) < vec(i + 1) ENTONCES
                        aux = vec(i)
                        vec(i) = vec(i + 1)
                        vec( i + 1) = aux
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
FIN
```

Cuando ordenamos en forma descendente, debemos invertir la pregunta en la condición de la instrucción SI ENTONCES   vec(i) < vec(i + 1).
El resto del algoritmo es igual a lo explicado para el ordenamiento ascendente.

Ejercicio 9

Dados un vector sueldo y un vector nombre$, cargarlos con los sueldos y nombres de 50 empleados y luego imprimir ambos, ordenados en forma ascendente por sueldo de empleado :

```
COMIENZO
sueldo(50) : nombre$(50)
PARA i = 1 A 50
```

```
            INGRESAR " Ingrese el nombre del empleado:", nombre$(i)
            INGRESAR " Ingrese el sueldo del empleado:", sueldo$(i)
PROXIMO
cota = 50
k = 1
HACER MIENTRAS k <> 0
            k = 0
            PARA i = 1 A cota - 1
                        SI sueldo(i) > sueldo(i + 1) ENTONCES
                                    aux = sueldo(i)
                                    sueldo(i) = sueldo(i + 1)
                                    sueldo( i + 1) = aux
                                    aux$ = nombre$(i)
                                    nombre$(i) = nombre$(i + 1)
                                    nombre$( i + 1) = aux$
                                    k = i
                        FIN SI
            PROXIMO
            cota = k
REPETIR
FIN
```

En este ejemplo se trabaja con dos vectores paralelos: sueldo que determina el ordenamiento y nombre$ que trabaja asociado al vector sueldo.

Ambos vectores se cargan paralelamente. Esto quiere decir que al nombre del empleado cargado en la posición 1 del vector nombre$, le va a corresponder el sueldo cargado en la posición 1 del vector sueldos.

El algoritmo de ordenamiento es el mismo que utilizamos en los ejemplos anteriores, con el agregado de tener que intercambiar el vector nombre$, cada vez que corresponda intercambiar el vector sueldo:

```
SI sueldo(i) > sueldo(i + 1) ENTONCES
            aux = sueldo(i)
            sueldo(i) = sueldo(i + 1)
            sueldo( i + 1) = aux
            aux$ = nombre$(i)
            nombre$(i) = nombre$(i + 1)
            nombre$( i + 1) = aux$
            k = i
FIN SI
```


La variable auxiliar utilizada debe ser alfanumérica como nombre$: aux$.

RESUMEN

Estructura de datos: es el conjunto de datos, organizados bajo un nombre común, que se pueden tratar como una sola unidad.

Las estructuras de datos, de acuerdo al lugar donde se almacenan, se clasifican en:

✗ Internas: se almacenan en la memoria de la computadora, se clasifican en vectores y matrices según el tipo.

✗ Externas: se almacenan en soportes externos como los discos; se denominan archivos.

✗ Un vector es un conjunto de datos del mismo tipo, numéricos o alfanuméricos, organizados bajo un mismo nombre y almacenados en la memoria de la computadora.

✗ Cada dato se denomina elemento del vector y está referenciado por una variable índice, que indica la posición de dicho elemento dentro del vector.

✗ Previo al uso de un vector, hay que reservar una zona de memoria para su uso, así como definir el número de elementos necesarios para acceder a cada elemento de la estructura. Al proceso de reservar una zona de la memoria para almacenar los datos del vector, se lo denomina dimensionar y se realiza de la siguiente forma:

Sintaxis: Nombre del vector(cantidad de elementos)

✗ La posición es la dirección del lugar que ocupa cada dato dentro del vector y está indicado por el índice.

✗ El elemento es el dato que hay almacenado en el vector, en esa posición.

✗ Los vectores, al residir en la memoria de la computadora, son de acceso directo, por lo que podemos situarnos directamente en la posición deseada.

✗ Los elementos de un vector se utilizan en forma análoga a la de cualquier variable.

✗ Los vectores que contienen datos alfanuméricos, deben escribirse con el símbolo $ al final: vector$(i).

Máximos y mínimos: al disponer de todos los datos invocando al vector, se define inicialmente a vmax, con el dato almacenado en el primer elemento: vmax = vec(i).

A continuación se recorre el vector desde el segundo elemento hasta el último, utilizando un ciclo PARA. Esta búsqueda permite encontrar el valor máximo buscado.

El algoritmo para calcular el máximo de un vector llamado vec(n) es

```
vec(n)
max = vec(1)
PARA i = 2 A n
          SI vec(i) > max ENTONCES max = vec(i)
PROXIMO
```

Donde n es la cantidad de elementos del vector.

El algoritmo para un mínimo sería:

```
vec(n)
min = vec(1)
PARA i = 2 A n
          SI vec(i) < min ENTONCES min = vec(i)
PROXIMO
```

En los procedimientos utilizados anteriormente, colocamos como restricción, que no podían existir máximos y mínimos múltiples; sólo un valor entre un conjunto podía estar asociado al resultado de la búsqueda.

Una gran ventaja en la utilización de vectores es la posibilidad de poder encontrar más de una variable asociada a un máximo o a un mínimo, que responda a la búsqueda solicitada. Esto es así cuando se trabaja con dos o más vectores asociados.

Si bien el valor máximo o mínimo sigue siendo uno sólo, al ser múltiple la

posibilidad de encontrar varios elementos en vectores asociados al de búsqueda, se denomina a estos valores asociados, hallados, como máximos y mínimos múltiples.

La búsqueda se hace en dos etapas: primero se determina el máximo en el vector de búsqueda se buscan todos los elementos asociados a este valor en los otros vectores.

Ordenamiento: El ordenamiento puede ser:

✓ **Ascendente:** los elementos están situados de menor a mayor.

✓ **Descendente:** los elementos están situados de mayor a menor.

Los valores repetidos, en caso de existir, quedan en posiciones contiguas.

Utilizamos el método del burbujeo que se basa en llevar el máximo del vector a la última posición. Para lograr esto se van tomando los elementos de a dos y se los va comparando e intercambiando hasta cumplir con el objetivo. Además para optimizar su funcionamiento se utiliza un switch que nos permite conocer si el vector está ordenado o no.

El algoritmo es el siguiente:

```
cota = cantidad de elementos a ordenar
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota –1
                SI vec(i) > vec(i + 1) ENTONCES
                        aux = vec(i)
                        vec(i) = vec(i + 1)
                        vec( i + 1) = aux
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
```

La variable k es un switch que cumple con las siguientes condiciones:

Si k = 0, implica que el vector está ordenado; si k <> 0 el vector está desordenado.

Al comienzo del ejercicio suponemos que el vector está desordenado, lo

que indicamos colocando k en 1 (k podría haber sido igual a cualquier valor distinto de cero).

Esto hace que nos aseguremos la entrada en el ciclo HACER MIENTRAS al preguntar por k <> 0.

Una vez dentro del ciclo se coloca a k en 0. De esta manera estamos suponiendo que una vez que se ejecuten las instrucciones de ciclo HACER MIENTRAS, el vector quedará ordenado.

A continuación tenemos un ciclo PARA de valor inicial igual a uno y de valor final a cota-1 (PARA i = 1 A cota-1), lo cual coincide con a cantidad de elementos, menos uno, que tiene el vector.

Esto está relacionado con la primer instrucción dentro del ciclo PARA:

SI vec(i) > vec(i + 1) ENTONCES , que compara el primer elemento con el segundo, el segundo con el tercero, el tercero con el cuarto, etc, hasta llegar a comparar los dos últimos elementos. Si el ciclo fuera de 1 a cota, la última comparación la haría entre el último elemento (i = cota) y un elemento inexistente (i + 1 = cota + 1). Esto explica el motivo de recorrer hasta cota-1.

Cada vez que la condición es verdadera, hay que hacer un intercambio de valores:

```
aux = vec(i)
vec(i) = vec(i + 1)
vec( i + 1) = aux
```

Como se puede apreciar se utiliza una variable auxiliar aux, para no perder ningún dato al copiar de vec(i) a vec(i+1) y viceversa

Luego aparece nuevamente una asignación a la variable k, esta vez del índice i:

k = i

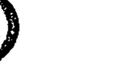
El valor de i en el momento de la asignación corresponde a la posición de los últimos elementos intercambiados, lo que tiene un doble significado:

✗ Cargar k con un valor distinto de cero para volver a ejecutar el ciclo HACER MIENTRAS, pues el hecho de haber realizado un intercambio condiciona a que se ejecute el ciclo al menos una vez más para verificar si el vector quedó ordenado.

✗ Transferir fuera del ciclo PARA, el valor de k hacia cota (cota = k), para no tener que volver a comparar nuevamente desde el primer elemento hasta

el último, sino hasta el último intercambiado. Se supone que el resto del vector no tuvo intercambios porque está ordenado.

El vector estará ordenado cuando al ejecutar íntegramente el ciclo PARA, nunca se cumpla la condición verificada por la instrucción SI ENTONCES. Si eso sucede no se asignará i a k, permaneciendo esta última en cero, valor cargado en k luego de ingresar en el ciclo HACER MIENTRAS. De esta forma no se cumplirá la condición k <> 0 y finalizará el proceso de ordenamiento.


Ejercicios Propuestos

1. Cargar un vector de 45 elementos e imprimir el valor de cada uno de estos.

2. Dados 20 números, cargarlos en un vector y hallar e imprimir:

a) la suma de los elementos.
b) la cantidad de elementos del vector iguales a 1.

3. Cargar un vector de n elementos y luego imprimir:

a) Los elementos pares.
b) La suma de los elementos.
c) El promedio de los elementos.
d) El porcentaje de elementos positivos.

4. Dados los sueldos y edades de r empleados de una empresa, se pide cargarlos en vectores y luego imprimir:

a) Sueldo promedio.
b) Sueldo promedio de los empleados que tengan entre 18 y 20 años.
c) Edad promedio.
d) Cantidad de empleados con sueldo mayor al sueldo promedio.
e) Cantidad de empleados con edad menor a la edad promedio.

5. Dadas n notas y edades de alumnos, se pide cargarlas en vectores y luego imprimir:

a) Cantidad de alumnos aprobados.
b) Cantidad de alumnos aplazados.
c) Edad promedio.
d) Nota promedio de los alumnos mayores a 15 años.

6. Se tiene un vector cargado con 100 números. Se pide hallar e imprimir,

6. Se tiene un vector cargado con 100 números. Se pide hallar e imprimir,
la cantidad de números positivos, negativos y ceros.

7. Se deben cargar en un vector los tiempos de clasificación de 60 autos. Los autos se identifican con números correlativos del ˙ al 60. Se pide imprimir:

a) Número del auto que clasificó primero.
b) Peor tiempo de clasificación.

8. Dado el vector tiempos del ejercicio 7, ordenarlo en forma ascendente. Se deberá imprimir de la siguiente forma:

Número de auto	Tiempo



Subprogramas

6.1 INTRODUCCIÓN

Los programas realizados hasta ahora, están todos desarrollados dentro de un único programa denominado **programa principal.**

En el capítulo 1, cuando hablamos de las características de los algoritmos, hicimos referencia al concepto de modularidad, que consiste en estrucurar el programa principal en módulos más pequeños llamados **subprogramas.**

Un subprograma es un conjunto de instrucciones de un programa que llevan a cabo una determinada tarea y que puede ejecutarse desde distintos puntos del programa principal.

Al finalizar la ejecución del subprograma se regresa al punto de partida del programa principal, continuando la secuencia de ejecución de ese.

La estructura de un subprograma es básicamente la de cualquier programa, con las diferencias lógicas en el comienzo y el fin.

Un subprograma puede a su vez estar compuesto por varios subprogramas. Están descriptos fuera del programa principal.

La estructura es la siguiente:

```
COMIENZO
NOMBRESUBPROGRAMA 1
NOMBRESUBPROGRAMA 2
NOMBRESUBPROGRAMA 3
.................................
NOMBRESUBPROGRAMA N
FIN

NOMBRESUBPROGRAMA 1:

Instrucciones
```

RETORNAR
NOMBRESUBPROGRAMA 2:

Instrucciones

RETORNAR
NOMBRESUBPROGRAMA 3:

Instrucciones

RETORNAR
....
....
NOMBRESUBPROGRAMA N:

Instrucciones

RETORNAR

Los nombres de subprogramas los escribiremos en mayúscula, para una mayor legibilidad del programa. Pueden utilizarse letras, números y guiones pero no deben dejarse espacios en blanco y siempre el primer carácter debe ser una letra.

En la llamada al subprograma dentro del programa principal, se coloca sólo el nombre. En el subprograma propiamente dicho, acompañaremos el nombre con dos puntos (:) al final y luego de todos las instrucciones indicamos el regreso al programa principal con la palabra RETORNAR.

La cantidad de subprogramas que habrá dentro de cada programa será determinada por el programador en función de la complejidad del ejercicio.

El objetivo a cumplir por los subprogramas es el de conseguir una mayor estructuración del programa, facilitando su construcción y simplificando al máximo las futuras modificaciones del programa.

Una gran ventaja de utilizar subprogramas , es la de evitar repeticiones de instrucciones dentro de un programa; estas instrucciones se escriben en un subprograma el cual es llamado y ejecutado las veces que haga falta.

De esta manera las instrucciones están escritas una sola vez, ocupando menos memoria en el almacenamiento del programa.

6.2 EJERCICIOS DE APLICACIÓN

Vamos a desarrollar un conjunto de ejercicios que nos permitan visualizar por un lado los conceptos vistos hasta ahora, y además comenzar a trabajar dividiendo el programa en un conjunto de subprogramas.

Ejercicio 1

Una editorial de libros posee 12 libros diferentes numerados del 1 al 12. Cuenta con una planilla en la que figuran los siguientes datos:

- Número de libro.
- Título.
- Autor.
- Costo.
- Precio de venta.
- Cantidad de ejemplares vendidos.

Se pide calcular e imprimir:

a) Título del libro con mayor cantidad de ejemplares vendidos.
b) Autor y título del libro con menor costo.
c) Facturación total de la editorial.
d) Ganancia de la editorial.

```
COMIENZO
INICIO
CARGA
MAXIMO
MINIMO
FACTURACION
GANANCIAS
IMPRESION
FIN

INICIO:
numero(12) : titulo$(12) : autor$(12) : costo(12) : precio(12) : cantidad(12)
RETORNAR
```

```
CARGA:
PARA i = 1 A 12
        INGRESAR "Número de libro:", numero(i)
        INGRESAR "Título:", titulo$(i)
        INGRESAR "Autor:", autor$(i)
        INGRESAR "Costo:", costo(i)
        INGRESAR "Precio de venta:", precio(i)
        INGRESAR "Cantidad de ejemplares vendidos:", cantidad(i)
PROXIMO
RETORNAR

MAXIMO:
max = cantidad(1)
PARA i = 2 A 12
        SI cantidad(i) > max ENTONCES max = cantidad(i)
PROXIMO
RETORNAR

MINIMO:
min = costo(1)
PARA i = 2 A 12
        SI costo(i) < min ENTONCES min = costo(i)
PROXIMO
RETORNAR

FACTURACION:
fact = 0
PARA i = 1 A 12
        fact = fact + precio(i)
PROXIMO
RETORNAR

GANANCIAS:
PARA i = 1 A 12
        sumcosto = sumcosto + costo(i)
PROXIMO
ganancia = fact – sumcosto
RETORNAR
```



```
IMPRESION:
IMPRIMIR "Libro de mayor cantidad de ejemplares vendidos:"
PARA i = 1 A 12
        SI cantidad(i) = max ENTONCES IMPRIMIR titulo$(i)
PROXIMO
IMPRIMIR "Libro de menor costo:"
PARA i = 1 A 12
        SI costo(i) = min ENTONCES IMPRIMIR autor$(i), titulo$(i)
PROXIMO
IMPRIMIR"Facturación total:", fact,"$"
IMPRIMIR "Ganancias:", ganancia,"$"
RETORNAR
```

Se dividió el programa en siete subprogramas: INICIO, CARGA, MAXIMO, MINIMO, FACTURACION, GANANCIAS e IMPRESION; recordar que al ser nombres de subprogramas no llevan acento al igual que las variables.

En INICIO se dimensionaron los 6 vectores para cargarlos luego en CARGA.

Los cálculos fueron realizados en los cuatros subprogramas siguientes: MAXIMO, MINIMO, FACTURACION Y GANANCIAS.

Los resultados se muestran todos en el subprograma IMPRESIÓN.

Ejercicio 2

Una empresa maneja sus ventas por medio de corredores: cada uno de ellos cobra un porcentaje sobre lo vendido.

Además el corredor que vende más de 5000$ recibe un premio de 2000$ sin descuento.

Los corredores son 7 y los porcentajes de comisión son:

1 y 2:	***10%***
3 y 5:	***5%***
4:	***7%***
6 y 7:	***3%***

Para cada corredor se conoce además el nombre y el apellido.
Se desea calcular e imprimir:

a) Ventas totales de la empresa.

b) Ventas por corredor.
c) Comisión en pesos de cada corredor.
d) Cantidad de pedidos por corredor.
e) Porcentaje de ventas de cada corredor sobre el total.
f) Número de corredor de mayor venta.
g) Nombre del corredor con menor comisión.

Los datos son:
Número de pedido.
Número de corredor.
Monto.

El ingreso de datos finaliza con un número de pedido igual a cero.

```
COMIENZO
INICIO
CARGA
COMISION
PORCENTAJE
MAXIMO
MINIMO
IMPRESIÓN
FIN

INICIO:
venta(7) : pedido(7) : compor(7) : porcent(7) : comision(7) : nombre$(7)
compor(1) = 0.1
compor(2) = 0.2
compor(3) = 0.05
compor(4) = 0.07
compor(5) = 0.05
compor(6) = 0.03
compor(7) = 0.03
PARA i = 1 A 7
INGRESAR "Nombre:", nombre$(i)
PROXIMO
        toventa = 0
PARA i = 1 A 7
        venta(i) = 0
```



```
        pedido(i) = 0
PROXIMO
RETORNAR

CARGA:
INGRESAR "Número de pedido:", np
HACER MIENTRAS np <> 0
        INGRESAR "Número de corredor:", nc
        INGRESAR "Monto de la venta:", mv
        totventa = totventa + mv
        venta(nc) = venta (nc) + mv
        pedido(nc) = pedido(nc) + 1
        INGRESAR "Número de pedido:", np
REPETIR
RETORNAR

COMISION:
PARA i = 1 A 7
        comision(i) = venta(i) * compor(i)
        SI venta(i) > 5000 ENTONCES comision(i) = comision(i) + 2000
PROXIMO
RETORNAR

PORCENTAJE:
PARA i = 1 A 7
        porcent(i) = venta(i) / totventa * 100
PROXIMO
RETORNAR

MAXIMO:
max = venta(1)
PARA i = 2 A 7
        SI venta(i) > max ENTONCES max = venta(i)
PROXIMO
RETORNAR

MINIMO:
min = comision(1)
PARA i = 2 A 7
```

```
        SI comision(i) < min ENTONCES min = comision(i)
PROXIMO
RETORNAR

IMPRESION:
IMPRIMIR "Ventas totales de la empresa:", totventa,"$"
IMPRIMIR "Ventas por corredor:"
PARA i = 1 A 7
        IMPRIMIR "Corredor:", i, "Ventas:", venta(i), "$"
PROXIMO
IMPRIMIR "Pedidos por corredor:"
PARA i = 1 A 7
        IMPRIMIR "Corredor:", i, "Pedidos:", pedido(i)
PROXIMO
IMPRIMIR "Comisión por corredor:"
PARA i = 1 A 7
        IMPRIMIR "Corredor:", i, "Comisión:", comision(i), "$"
PROXIMO
IMPRIMIR "Porcentaje de ventas de cada corredor:"
PARA i = 1 A 7
        IMPRIMIR "Corredor:", i, "Porcentaje:", porcent(i), "%"
PROXIMO
IMPRIMIR "Corredores de mayor venta:"
PARA i = 1 A 7
        SI venta(i) = max ENTONCES IMPRIMIR "Corredor:", i
PROXIMO
IMPRIMIR "Nombres de los corredores con menor comisión:"
PARA i = 1 A 7
        SI comision(i) = min ENTONCES IMPRIMIR nombre$(i)
PROXIMO
RETORNAR
```

Se dividió el programa en 7 subprogramas, nuevamente INICIO y CARGA al comienzo, luego los distintos procesos de cálculo: COMISION, PORCENTAJE, MAXIMO y MINIMO; finalizando con el subprograma de impresión.

Se dimensionaron 6 vectores de los cuales venta() y pedido() se colocaron en cero por ser un acumulador y un contador respectivamente. Es interesante ver como se inicializó compor(), con los distintos porcentajes de comisión de acuerdo al número de corredor.

En la carga, no sólo se ingresaron los datos, además se cargaron los vectores venta() y pedido().

La comisión es un vector que se carga con el producto entre dos vectores: venta() y compor().

El porcentaje como siempre está dado por el cociente entre el valor individual y el total por 100.

Los subprogramas de máximo, mínimo e impresión son similares a los vistos en ejercicios anteriores.

Ejercicio 3

Una empresa de informática tiene 100 empleados. Cada uno de ellos pertenece a una categoría que se le asigna cuando ingresa a trabajar en la empresa.

Las categorías son cinco:

1- Analista Senior.
2- Analista Junior.
3- Programador Senior.
4- Programador Junior.
5- Operador.

Los sueldos son:

1- 2500$
2 2000$
3- 1500$
4- 1200$
5- 800$

El empleado puede trabajar en dichas categorías en alguno de los tres departamentos que posee la empresa. Los sueldos son fijos para cada categoría.

A fin de mes se liquidan los sueldos en una planilla donde figura:

- Nombre del empleado.
- Categoría.
- Departamento (1, 2 y 3).

Se pide calcular e imprimir:

a) Cantidad de empleados por categoría.
b) Sueldos totales de cada categoría.
c) Cantidad de empleados por departamento.
d) Nombre de la categoría que tenga más empleados.
e) Número de departamento que tenga la mínima cantidad de empleados.
f) Imprimir en forma ordenada ascendente por sueldos de cada categoría un listado con los siguientes datos:

- Número de categoría.
- Nombre de la categoría.
- Sueldo total.
- Cantidad de empleados.

```
COMIENZO
INICIO
CARGA
MAXIMO
MINIMO
ORDEN
IMPRESION
FIN

INICIO:
empcat(5) : sdocat(5) : empdep(3) : nombre$(5) : sueldo(5) : numero(5)
nombre$(1) = "Analista Senior"
nombre$(2) = "Analista Junior"
nombre$(3) = "Programador Senior"
nombre$(4) = "Programador Junior"
nombre$(5) = "Operador"
sueldo(1) = 2500
sueldo(2) = 2000
sueldo(3) = 1500
sueldo(4) = 1200
sueldo(5) = 800
PARA i = 1 A 5
        numero(i) = i
        empcat(i) = 0
        sdocat(i) = 0
```



```
PROXIMO
PARA i = 1 A 3
        empdep(i) = 0
PROXIMO
RETORNAR

CARGA:
PARA i = 1 A 100
        INGRESAR "Categoría:", cat
        INGRESAR "Número de departamento:", dep
        empcat(cat) = empcat(cat) + 1
        empdep(dep) = empdep(dep) + 1
        sdocat(cat) = sdocat(cat) + sueldo(cat)
PROXIMO
RETORNAR

MAXIMO:
max = empcat(1)
PARA i = 2 A 5
        SI empcat(i) > max ENTONCES max = empcat(i)
PROXIMO
RETORNAR

MINIMO:
min = empdep(1)
PARA i = 2 A 3
        SI empdep(i) < min ENTONCES min = empdep(i)
PROXIMO
RETORNAR

ORDEN:
cota = 5
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota - 1
                SI sdocat(i) > sdocat(i + 1) ENTONCES
                        aux = sdocat (i)
                        sdocat (i) = sdocat (i + 1)
```

```
                        sdocat ( i + 1) = aux
                        aux = numero (i)
                        numero (i) = numero (i + 1)
                        numero ( i + 1) = aux
                        aux$ = nombre$ (i)
                        nombre$ (i) = nombre$ (i + 1)
                        nombre$ ( i + 1) = aux$
                        aux = empcat(i)
                        empcat (i) = empcat (i + 1)
                        empcat ( i + 1) = aux

                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
RETORNAR

IMPRESION:
IMPRIMIR "Cantidad de empleados por departamento:"
PARA i = 1 A 3
        IMPRIMIR "Departamento:", i, "Empleados:", empdep(i)
PROXIMO
IMPRIMIR "Categoria con más empleados:"
PARA i = 1 A 5
        SI empcat(i) = max ENTONCES IMPRIMIR nombre$(i)
PROXIMO
IMPRIMIR "Departamentos con mínima cantidad de empleados:"
PARA i = 1 A 3
        SI empdep(i) = min ENTONCES IMPRIMIR "Departamento ", i
PROXIMO
IMPRIMIR "Sueldos por categoría:"
PARA i = 1 A 5
        IMPRIMIR numero(i)
        IMPRIMIR nombre$(i)
        IMPRIMIR "sueldo:", sdocat(i), "$"
        IMPRIMIR "Cantidad de empleados:", empcat(i)
PROXIMO
RETORNAR
```


Se dividió el programa en 6 subprogramas: INICIO, CARGA, MAXIMO, MINIMO, ORDEN e IMPRESION.

Trabajamos con 6 vectores, dos de los cuales son contadores: empcat() y empdep(); y un acumulador sdocat().

En este caso tenemos como novedad la utilización del subprograma ORDEN que lleva a cabo el ordenamiento. El vector que ordena es el de sueldos por categoría sdocat() y trabajan asociados a él y por lo tanto deben intercambiar sus elementos, los vectores: numero(), nombre$(), empcat().

Ejercicio 4

Una aerolínea comercial vende pasajes a cuatro destinos del exterior del país:

1- París.
2- Roma.
3- Madrid.
4- Londres.

Los aviones tienen tres clases:

1- Primera:	1800$
2- Negocios.	1600$
3- Turista.	900$

Si el destino es Roma y viaja en primera, se le descuenta el 10% del precio total.

Se pide calcular e imprimir:

a) Cantidad de pasajes vendidos a cada destino.
b) Recaudación por cada destino.
c) Porcentaje de ventas en cantidad de pasajes de cada destino.
d) Cantidad de pasajes vendidos por clase.
e) Nombre de la clase con mayor recaudación.
f) Impuestos pagados por la empresa (15% de la recaudación total).
g) Número de destino con menor cantidad de pasajes vendidos.

Los items a, b y c se deberán imprimir ordenados de manera ascendente por recaudación de cada destino de la siguiente forma:

- Número de destino
- Recaudcción por destino.
- Porcentcje de ventcs de cada destino.
- Cantidad de pasajes vendidos por destino.

```
COMIENZO
NICIO
CARGA
PORCENTAJE
MAXIMO
MINIMO
IMPUESTOS
ORDEN
IMPRESION
FIN

INICIO:
numero(4) : recdest(4) : pasdest(4) : porcent(4)
pasclase(3) : recclase(3) : nombre$(3)
PARA i = 1 A 4
        rumero(i) = i
        recdest(i) = 0
        pasdest(i) = 0
PROXIMO
PARA i = 1 A 3
        recclase(i) = 0
        pasclase(i) = 0
PROXIMO
nombre$(1) = "Primera"
nombre$(2) = " Negocios"
nombre$(3) = " Turista"
RETORNAR

CARGA:
INGRESAR "Número de pasaje:", np
HACER MIENTRAS np <> 0
        INGRESAR "Número de destino:", nd
        INGRESAR "Número de clase:", nc
        SELECCIONAR CASO nc
```



```
                CASO = 1
                        SI nd = 2 ENTONCES vp = 1800 *
                        0.9 SINO vp = 1800
                CASO = 2
                        vp = 1600
                CASO = 3
                        vp = 900
        FIN SELECCIONAR
        pasdest(nd) = pasdest(nd) + 1
        recdest(nd) = recdest(nd) + vp
        pasclase(nc) = pasclase(nc) + 1
        recclase(nc) = recclase(nc) + vp
REPETIR
RETORNAR

PORCENTAJE:
totpas = 0
PARA i = 1 A 4
        totpas = totpas + pasdest(i)
PROXIMO
PARA i = 1 A 4
        porcent(i) = pasdest(i) / totpas * 100
PROXIMO
RETORNAR

MAXIMO:
max = recclase(1)
PARA i = 2 A 3
        SI recclase(i) > max ENTONCES max = recclase(i)
PROXIMO
RETORNAR

MINIMO
min = pasdest(1)
PARA i = 2 A 4
        SI pasdest(i) < min ENTONCES min = pasdest(i)
PROXIMO
RETORNAR
```

```
IMPUESTOS:
totrec = 0
PARA i = 1 A 3
        totrec = totrec + recclase(i)
PROXIMO
impuesto = totrec * 0.15
RETORNAR

ORDEN:
cota = 4
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                SI recdest(i) > recdest (i + 1) ENTONCES
                        aux = recdest (i)
                        recdest (i) = recdest (i + 1)
                        recdest ( i + 1) = aux
                        aux = numero (i)
                        numero (i) = numero (i + 1)
                        numero ( i + 1) = aux
                        aux = porcent(i)
                        porcent (i) = porcent (i + 1)
                        porcent ( i + 1) = aux
                        aux = pasdest(i)
                        pasdest (i) = pasdest(i + 1)
                        pasdest ( i + 1) = aux
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
RETORNAR

IMPRESION:
PARA i = 1 A 4
        IMPRIMIR "Número de destino:", numero(i)
        IMPRIMIR "Recaudación:", recdest(i), "$"
        IMPRIMIR "Porcentaje de ventas:", porcent(i), "%"
```



```
        IMPRIMIR "Cantidad de pasajes vendidos:", pasdest(i)
PROXIMO
IMPRIMIR "Ventas por clase:"
PARA i = 1 A 3
        IMPRIMIR "Clase:", nombre$(i)
        IMPRIMIR "Pasajes vendidos:", pasclase(i)
PROXIMO
IMPRIMIR "Clase de mayor recaudación:"
PARA i = 1 A 3
        SI recclase(i) = max ENTONCES IMPRIMIR nombre$(i)
PROXIMO
IMPRIMIR "Número de destino de menor recaudación:"
PARA i = 1 A 4
        SI pasdest(i) = min ENTONCES IMPRIMIR numero(i)
PROXIMO
IMPRIMIR "Impuestos a pagar:", impuesto, "$'
RETORNAR
```

El programa está dividido en 8 subprogramas: INICIO, CARGA, PORCENTAJE, MAXIMO, MINIMO, IMPUESTOS, ORDEN e IMPRESION.

Todos estos algoritmos ya fueron analizados con distintas variantes, en ejercicios anteriores.

Se utilizaron 7 vectores, de los cuales 2 son acumuladores: recdest() y recclase(); y dos contadores: pasdest() y pasclase().

Es interesante observar como en la carga se utilizó un instrucción de SELECCIONAR CASO, para determinar el valor de pasaje (vp) a acumular en el cálculo y la carga de los vectores de recaudación por destino y por clase.

Los subprogramas de cálculo de máximo, minimo y porcentaje no difieren de los algoritmos clásicos estudiados.

El subprograma de impuesto es un cálculo porcentual directo, del total recaudado.

En el ordenamiento tenemos cuatro vectores numéricos intercambiando elementos, en función del vector de recaudación por destino recdest().

RESUMEN

Un subprograma es un conjunto de instrucciones de un programa que llevar a cabo una determinada tarea y que puede ejecutarse desde distintos puntos del programa principal.

Al finalizar la ejecución del subprograma se regresa al punto de partida del programa principal, continuando la secuencia de ejecución de este.

La estructura de un subprograma es básicamente la de cualquier programa, con las diferencias lógicas en el comienzo y el fin.

Un subprograma puede a su vez estar compuesto por varios subprogramas.

Están descriptos fuera del programa principal.

La estructura es la siguiente:

```
COMIENZO
NOMBRESUBPROGRAMA 1
NOMBRESUBPROGRAMA 2
NOMBRESUBPROGRAMA 3
...................................
NOMBRESUBPROGRAMA N
FIN

NOMBRESUBPROGRAMA 1:

Instrucciones

RETORNAR
NOMBRESUBPROGRAMA 2:

Instrucciones

RETORNAR
NOMBRESUBPROGRAMA 3:
```



```
Instrucciones

RETORNAR
.....................................
.........................................
NOMBRESUBPROGRAMA N:

Instrucciones

RETORNAR
```

El objetivo a cumplir por los subprogramas es el de conseguir una mayor estructuración del programa, facilitando su construcción y simplificando al máximo las futuras modificaciones del programa.

Una gran ventaja de utilizar subprogramas , es la de evitar repeticiones de instrucciones dentro de un programa; estas instrucciones se escriben en un subprograma el cual es llamado y ejecutado las veces que haga falta.

De esta manera las instrucciones están escritas una sola vez, ocupando menos memoria en el almacenamiento del programa.



Ejercicios Propuestos

1. La administración del estado posee 1200 empleados. Cada uno de ellos posee una categoría. Las categorías son tres.
Los diferentes empleados se distribuyen en las siguientes dependencias

1- Trabajo.
2- Educación.
3- Economía.
4- Interior.

Los sueldos por categoría son:

1- 600
2- 500
3- 400

El empleado puede trabajar en dichas categorías en alguna de las cuatro dependencias.
Los sueldos son fijos para cada categoría:
A fin de mes se liquidan los sueldos en una planilla donde figura:

- Nombre del empleado.
- Categoría.
- Dependencia.

Se pide calcular e imprimir:

a) Cantidad de empleados por dependencia.
b) Sueldos totales de cada categoría.
c) Cantidad de empleados por dependencia.
d) Nombre de la categoría que tenga menos empleados.
e) Número de dependencia que halla pagado menos sueldos en pesos.
f) Imprimir en forma ordenada ascendente por sueldos de cada depen-

dencia, un listado con los siguientes datos:

- Número de dependencia.
- Nombre de la dependencia.
- Sueldo total de la dependencia.
- Cantidad de empleados de la dependencia.

2. Una empresa transportadora de caudales posee 4 camiones blindados para realizar sus viajes.
Cada camión puede transportar 3 tipos diferentes de valores:

1- Dólares.
2- Pesos.
3- Monedas de Oro.

Cada camión transporte solamente uno de los tres tipos de valores en un viaje.
Dependiendo del tipo de valor que transporta el camión, es el precio que cobra:

1- 300 por viaje.
2- 200 por viaje.
3- 500 por viaje.

Estos valores son independientes del camión que realiza el viaje.
A fin de mes la empresa efectúa una planilla con los siguientes datos:

- Nombre del cliente.
- Tipo de valor.
- Camión que realizó el viaje.

Son 1500 viajes. Se pide calcular e imprimir:

a) Cantidad de viajes de cada camión.
b) Recaudación de cada tipo de valor.
c) Recaudación por cada camión.
d) Nombre del valor que se haya transportado más veces.
e) Número de camión que haya recaudado menos dinero.
f) Imprimir ordenado de manera ascendente por recaudación de tipo de

valor, los siguientes datos:

- Número de valor.
- Nombre del valor.
- Recaudación por valor.
- Cantidad de veces que se transportó cada valor.

3. Una empresa enlatadora de frutas y verduras tiene 4 centros para enlatar y vender sus productos:

1- Capital.
2- Mendoza.
3- Córdoba.
4- Tucumán.

En cada centro se enlatan 5 productos diferentes.
Los precios de los productos son:

1- 0.80
2- 1.00
3- 0.35
4- 0.50
5- 0.70

A fin de mes la empresa confecciona una planilla con los datos de las ventas:

- Número de cliente.
- Número de centro.
- Número de producto.
- Cantidad de unidades.

Son 200 ventas registradas en el mes
Se pide calcular e imprimir:

c) Cantidad de ventas de cada centro.
b) Recaudación de cada producto.
c) Recaudación de cada centro.
d) Nombre del centro que recaudó menos.
e) Número del producto que se vendió más en cantidad de unidades.

f) Imprimir ordenado en forma ascendente por recaudación de cada centro, los siguientes datos:

- Número del centro.
- Nombre del centro.
- Recaudación del centro.

4. Una empresa de combustibles tiene 20 estaciones de servicio. En cada una de ellas se vende:

1- Nafta común	0.85 $ por litro.
2- Nafta especial	1.10 $ por litro.
3- Gas Oil	0.50 $ por litro.

Cada vez que un automovilista carga alguno de los tres combustibles, se emite una boleta con los siguientes:

- Número de boleta.
- Número de estación (del 1 al 20).
- Tipo de combustible (1, 2 ó 3).
- Cantidad de litros.

A fin de mes se procesan las ventas de todas las estaciones, teniendo en cuenta cada una de las boletas emitidas. El ingreso de datos finaliza con un número de boleta igual a cero.
Los nombres de las estaciones se deberán cargar en un vector en memoria.
Se pide calcular e imprimir:

a) Cantidad de litros vendidos en cada estación.
b) Recaudación de cada estación.
c) Cantidad de boletas emitidas por cada estación.
d) Total de litros vendidos para cada tipo de combustible.
e) Número de estación que recaudó más.
f) Número de boleta con mayor importe.
g) Impuestos pagados (10% de la recaudación total).

Los items a, b y c se deberán imprimir ordenados en forma ascendente, por recaudación por estación, de la siguiente manera:

- Número de estación.
- Recaudación por estación.
- Cantidad de boletas por estación.
- Centidad de litros por estación.

5. Una empresa envasadora de productos comestibles tiene 10 sucursales. En cada una de ellas se vende:

1- Arroz	0.20$ por kilo.
2- Yerba	0.50$ por kilo.
3- Azucar	0.30$ por kilo.
4- Cacao	0.20$ por kilo.

Cada vez que un mayorista compra alguno de los cuatro productos comestibles, se emite una boleta con los siguientes datos:

- Número de boleta.
- Número de sucursal (del 1 al 10).
- Número de producto (1, 2 3 ó 4).
- Cantidad de kilos.

A fin de mes se procesan las ventas de todas las sucursales, teniendo en cuenta cada una de las boletas emitidas. El ingreso de datos finaliza con un número de boleta igual a cero.

Los nombres de las sucursales se deberán cargar en un vector en memoria, al igual que los nombres de los productos.

Se pide calcular e imprimir:

a) Cantidad de kilos vendidos en cada sucursal.
b) Recaudación de cada sucursal.
c) Cantidad de boletas emitidas por cada sucursal.
d) Total de kilos vendidos para cada producto.
e) Nombre del producto que más recaudó.
f) Número de boleta con menor importe.

Los items a, b, y c se deberán imprimir ordenados en forma descendente por recaudación de cada sucursal de la siguiente manera:

- Número de sucursal.
- Recaudación por sucursal.
- Cantidad de boletas por sucursal
- Cantidad de litros por sucursal.

6. Una empresa de micros vende pasajes a tres destinos del interior del país:

- Córdoba.
- Corrientes.
- Misiones.

Siendo un proyecto piloto en materia de transporte automotor, los micros tienen tres clases:

1- Común	45$
2- Turista	50$
3- Pullman	55$

El costo del pasaje es el mismo para los tres destinos.

Al pasajero se le entrega un ticket donde se consignan los siguientes datos:

- Número de ticket.
- Número de destino.
- Número de clase .
- Cantidad de pasajes.

Si una persona compra más de 10 pasajes juntos, se le descuenta el 10% del precio total.

A fin de mes se procesan las ventas de todos los destinos, teniendo en cuenta cada uno de los tickets emitidos. El ingreso de datos finaliza con un número de ticket igual a cero.

Se pide calcular e imprimir:

a) Cantidad de pasajes vendidos a cada destino.
b) Recaudación por cada destino.
c) Porcentaje de ventas, en cantidad de pasajes, de cada destino.

d) Cantidad de pasajes vendidos por clase.
e) Número de ticket con mayor importe en una venta.
f) Destino de menor recaudación.

Los items a, b, y c se deberán imprimir ordenados de manera ascendente por recaudación de cada destino de la siguiente forma:

- Número de destino.
- Recaudación por destino.
- Porcentaje de ventas de cada destino.
- Cantidad de pasajes vendidos por destino.

7 Una entidad de beneficencia tiene autorizados 30 puestos de venta de panchos en las distintas estaciones de ferrocarriles. En cada una de ellas se vende:

- Panchos 1.0$.
- Hamburguesa 1.5$.
- Gaseosa 0.8$.

Cada vez que una persona compra alguno de los tres artículos, se emite un ticket con los siguientes datos:

- Número de ticket.
- Número de puesto.
- Total gastado.

A fin de mes se procesan las ventas de todos los puestos, teniendo en cuenta cada uno de los tickets emitidos. El ingreso de datos finaliza con un número de ticket igual a cero.

Los nombres de las estaciones se deberán cargar en un vector en memoria.

Se pide calcular e imprimir:

a) Recaudación de cada puesto.
b) Cantidad de tickets emitidos por cada puesto.
c) Cantidad de tickets con importe mayor a 10$.
d) Número de puesto de mayor recaudación.
e) Número de ticket con mayor importe.

f) Números de puesto con recaudaciones menores a 1500$

Los items c y b se deberán imprimir ordenados en forma descendente por recaudación de cada puesto de la siguiente manera:

- Número de puesto.
- Recaudación por puesto.
- Cantidad de boletas por puesto.

capítulo 7

Matrices

7.1 INTRODUCCIÓN

Una matriz es un conjunto de datos homogéneos organizados bajo un mismo nombre, cada uno de los cuales debe referenciarse por dos índices, **i** y **j**, también llamados subíndices, que identifican las filas y las columnas de la matriz **i** x **j**.

Ambos índices deben ser enteros mayores o iguales a cero.

Para dimensionar una matriz se necesitan los dos índices:

matriz (i , j) donde:

- i , es la cantidad de filas.
- j , la cantidad de columnas.
- matriz, el nombre de la matriz mediante el cual la computadora va a reconocer a ese conjunto de datos. Debe cumplir la misma normativa que una variable.

Los elementos de una matriz están organizados en filas y columnas como si fueran un grupo de j vectores de i elementos cada uno o viceversa.

Los elementos de una fila tienen todos ellos igual valor para el primer índice, i, mientras que los de una columna tienen igual valor para el segundo, j.

El hecho de asignar el primer índice a las filas y el segundo a las columnas, es sólo una cuestión de nomenclatura. No hay ningún impedimento en asignar i a las columnas y j a las filas, siempre y cuando seamos coherentes en todo el programa.

Ejemplo: la representación de una matriz mat, de 4 x 5, sería una matriz de cuatro filas y cinco columnas:

Columnas

FILAS	1	2	3	4	5
1	mat(1,1)	mat(1,2)	mat(1,3)	mat(1,4)	mat(1,5)
2	mat(2,1)	mat(2,2)	mat(2,3)	mat(2,4)	mat(2,5)
3	mat(3,1)	mat(3,2)	mat(3,3)	mat(3,4)	mat(3,5)
4	mat(4,1)	mat(4,2)	mat(4,3)	mat(4,4)	mat(4,5)

El número de elementos de la matriz es 4 x 5 = 20

Supongamos que tenemos una matriz nombre$ en la memoria de la computadora, con los nombres siguientes:

Columnas

FILAS	1	2	3	4	5
1	Mario				
2			Patricia		
3				Nicolás	
4		Florencia		Andrés	Elvira

De tal forma que:

nombre$(1,1) = Mario
nombre$(2,3) = Patricia
nombre$(3,4) = Nicolás
nombre$(4,2) = Florencia
nombre$(4,4) = Andrés
nombre$(4,5) = Elvira

Como las matrices están almacenadas en la memoria de la computadora, sus elementos son de acceso directo.

Podemos ir directamente a una posición determinada simplemente indicando sus coordenadas.

El tipo de elementos que se almacenan en las matrices es el mismo en todas sus posiciones, todos numéricos o todos alfanuméricos, pro nunca una mezcla de ambos.

Cuando se necesita almacenar información de distinto tipo, se deben utilizar vectores paralelos.

7.2 CARGA Y LECTURA DEL CONTENIDO DE UNA MATRIZ

Para cargar o leer una matriz se deben utilizar dos ciclos PARA anidados, recorriendo la matriz por filas y por columnas.

Por ser ciclos anidados hay que indicar la variable i o j, según corresponda, sobre la instrucción PROXIMO:

PROXIMO i o PROXIMO j

Supongamos una matriz mat, de f filas y c columnas. La forma de recorrerla es la siguiente:

- **Ingreso de datos**

```
PARA i = 1 A f
        PARA j = 1 A c
                    INGRESAR " Ingresar dato:", mat(i,j)
        PROXIMO j
PROXIMO i
```

Ejemplo: cargar en una matriz de 7 x 20, números enteros

```
PARA i = 1 A 7
        PARA j = 1 A 20
                    INGRESAR " Ingresar número:", numero(i,j)
        PROXIMO j
PROXIMO i
```

- **Inicialización**

```
PARA i = 1 A f
        PARA j = 1 A c
                    mat(i,j) = 0
        PROXIMO j
PROXIMO i
```

Ejemplo: poner a cero una matriz tiempo de 5 x 10

```
PARA i = 1 A 5
        PARA j = 1 A 10
                    tiempo(i,j) = 0
        PROXIMO j
PROXIMO i
```

- **Lectura para un cálculo**

```
PARA i = 1 A f
        PARA j = 1 A c
                    suma = suma + mat(i,j)
        PROXIMO j
PROXIMO i
```

Ejemplo: realizar la suma de todos los elementos de una matriz sueldo de 8 x 15

```
PARA i = 1 A 8
        PARA j = 1 A 15
                    suma = suma + sueldo(8,15)
        PROXIMO j
PROXIMO i
```

- **Lectura para impresión**

```
PARA i = 1 A f
```



```
        PARA j = 1 A c
                    IMPRIMIR" Resultado:", mat(i,j)
        PROXIMO j
PROXIMO i
```

Ejemplo: imprimir el contenido de una matriz notas de 40 x 9

```
PARA i = 1 A 40
        PARA j = 1 A 9
                    IMPRIMIR" Nota:", notas(i,j)
        PROXIMO j
PROXIMO i
```

7.3 MÁXIMOS Y MÍNIMOS

El proceso de determinación de un valor máximo o mínimo de una matriz, es muy similar a la búsqueda en vectores. Se inicializa la variable que contendrá el máximo o el mínimo, con el primer elemento de la matriz: (1,1). Luego se recorre nuevamente toda la matriz en la búsqueda de algún valor que supere o sea superado de acuerdo al tipo de valor que queremos encontrar: máximo o mínimo.

Cuando recorremos la matriz debemos lo hacemos desde i igual a 1 y j igual a 1. Esto significa volver a evaluar el elemento (1,1), pues a diferencia de la búsqueda en vectores, no hacerlo traería como consecuencia no tener en cuenta el resto de la primera fila.

A continuación podemos ver la sintaxis de los algoritmos para buscar el máximo y el mínimo en una matriz:

- **Búsqueda de un máximo**

```
max = mat(1,1)
PARA i = 1 A f
        PARA j = 1 A c
                    SI mat(i,j) > max ENTONCES max = mat(i,j)
        PROXIMO j
PROXIMO i
```

Ejemplo: buscar el peso máximo cargado en una matiz peso de 12 x 35.

```
max = peso(1,1)
PARA i = 1 A 12
        PARA j = 1 A 35
                SI peso(i,j) > max ENTONCES max = peso(i,j)
        PROXIMO j
PROXIMO i
```

- **Búsqueda de un mínimo**

```
min = mat(1,1)
PARA i = 1 A f
        PARA j = 1 A c
                SI mat(i,j) > min ENTONCES min = mat(i,j)
        PROXIMO j
PROXIMO i
```

Ejemplo: hallar la temperatura mínima cargada en una matriz temp de 4 x 7

```
min = temp(1,1)
PARA i = 1 A 4
        PARA j = 1 A 7
                SI temp(i,j) > min ENTONCES min = temp(i,j)
        PROXIMO j
PROXIMO i
```

7.4 ORDENAMIENTO DE MATRICES

Una matriz se ordena utilizando el mismo método que para un vector.
Al tener más de una columna, debemos decidir por cuál de ellas queremos ordenarla.

También debemos tener en cuenta que cada vez que se produzca un intercambio de los elementos de una matiz, afecta a todos los elementos que están situados en la misma fila.

Para intercambiar los elementos que están situados en la misma fila, pero en columnas distintas, al ser del mismo tipo (todos numéricos o todos

alfanuméricos), utilizamos un ciclo que recorra todas las columnas para esa misma fila y los intercambie.

A continuación analizaremos el siguiente ejemplo:

Ordenar una matriz mat(10,4), por la primera columna.

```
COMIENZO
mat(10,4)
cota = 10
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                SI mat(i,1) > mat(i + 1,1) ENTONCES
                        PARA j =1 A 4
                                aux = mat(i,j)
                                mat(i,j) = mat(i + 1,j)
                                mat(i + 1,j) = aux
                        PROXIMO
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
FIN
```

Para recorrer la matriz se necesitan dos ciclos PARA, uno para comparar los elementos en la columna 1: mat(i,1) > mat(i + 1,1) ; y otro para intercambiar los elementos de todas las columnas, cuando está desordenada la columna 1: PARA j =1 A 4 .

7.5 EJERCICIOS DE APLICACIÓN

Ejercicio 1

Generar una matriz mat de 5 x 10, introduciendo los valores por el teclado.

Se pide imprimir:

a) La matriz.
b) La suma de los elementos impares.
c) El elemento mat(3,5).

```
COMIENZO
mat(5,10)
suma = 0

PARA i = 1 A 5
        PARA j = 1 A 10
                INGRESAR " Ingresar dato:", mat(i,j)
        PROXIMO j
PROXIMO i

PARA i = 1 A 5
        PARA j = 1 A 10
                SI (-1) ^ mat(i,j) < 0 ENTONCES suma = suma + mat(i,j)
        PROXIMO j
PROXIMO i

PARA   = 1 A 5
        PARA j = 1 A 10
                IMPRIMIR " Elemento ", i, ", ", j, ":", mat(i,j)
        PROXIMO j
PROXIMO i

IMPRIMIR" La suma de elementos impares es:", suma
IMPRIMIR "El elemento mat(3,5) es igual a  ", mat(3,5)
FIN
```

En el ejercicio se dimensionó la matriz y se colocó a cero el acumulador utilizado para la suma de los elementos impares. Luego se cargó la matriz con los 50 valores.

La determinación de si un elemento es impar se llevó a cabo elevando −1 a el valor de dicho elemento, si el resultado es menor que cero, el elemento es impar:


```
SI (-1) ^ mat(i,j) < 0  ,  mat(i,j) es impar
```

Podemos observar que cuando queremos imprimir un elemento en particular, accedemos directamente a él, colocando el valor de los índices en esa posición:

```
IMPRIMIR "El elemento mat(3,5) es igual a  ", mat(3,5)
```

Ejercicio 2

Cargar una matriz mat(5,5), introduciendo los valores por teclado. Se pide imprimir:

a) La suma de los elementos de la diagonal principal.
b) La suma de todos los elementos.
c) La suma de la columna 5.

```
COMIENZO
INICIO
CARGA
DIAGONAL
TOTAL
COLUM-5
IMPRESION
FIN

INICIO:
mat(5,5)
sumadiag = 0
sumatot = 0
sumacol5 = 0
RETORNAR

CARGA:
PARA i = 1 A 5
        PARA j = 1 A 5
                INGRESAR " Ingresar dato:", mat(i,j)
        PROXIMO j
```

```
PROXIMO i
RETORNAR

DIAGONAL:
PARA i = 1 A 5
        PARA j = 1 A 5
                SI i = j ENTONCES sumadiag = sumadiag + mat(i,j)
        PROXIMO j
PROXIMO i
RETORNAR

TOTAL:
PARA i = 1 A 5
        PARA j = 1 A 5
                sumatot = sumatot + mat(i,j)
        PROXIMO j
PROXIMO i
RETORNAR

COLUM-5:
PARA i = 1 A 5
        sumacol5 = sumacol5 + mat(i,5)
PROXIMO i
RETORNAR

IMPRESION:
IMPRIMIR "La suma de la diagonal principal es:", sumadiag
IMPRIMIR" La suma total es:", sumatot
IMPRIMIR "La suma de a columna 5 es:", sumacol5
RETORNAR
```

En este caso dividimos el programa en 6 subprogramas.

Se trabajó con 3 acumuladores de suma.

La determinación de la suma de la diagonal se basa en considerar para la misma aquellos elementos que tienen iguales ambos índices i y j.

Se puede apreciar como se suma la columna 5 a partir de colocar un ciclo PARA que recorre sólo los elementos de la columna 5. Esto se logra manteniendo constante el valor de la columna a recorrer: mat(i,5).

Ejercicio 3

Generar una matriz de n filas y m columnas. Cargar la matriz con números enteros mayores a cero.

Se pide calcular e imprimir:

a) El valor máximo de toda la matriz.
b) El valor mínimo de toda la matriz.
c) La cantidad de veces que se repite el número 10.

```
COMIENZO
INICIO
CARGA
MAXIMO
MINIMO
VECES-10
IMPRESION
FIN

INICIO:
INGRESAR "Ingrese la cantidad de filas:", n
INGRESAR "Ingrese la cantidad de columnas", m
mat(n,m)
RETORNAR

CARGA:
PARA i = 1 A n
        PARA j = 1 A m
                INGRESAR "Ingrese un número:", mat(i,j)
        PROXIMO j
PROXIMO i
RETORNAR

MAXIMO:

max = mat(1,1)
PARA i = 1 A n
        PARA j = 1 A m
                SI mat(i,j) > max ENTONCES max = mat(i,j)
```

```
        PROXIMO j
PROXIMO i
RETORNAR

MINIMO:
min = mat(1,1)
PARA i = 1 A n
        PARA j = 1 A m
                SI mat(i,j) > min ENTONCES min = mat(i,j)
        PROXIMO j
PROXIMO i
RETORNAR

VECES-10:
contador = 0
PARA i = 1 A n
        PARA j = 1 A m
                SI mat(i,j) = 10 ENTONCES contador = contador + 1
        PROXIMO j
PROXIMO i
RETORNAR

IMPRESION:
IMPRIMIR "El valor máximo es:", max
IMPRIMIR "El valor mínimo es:", min
IMPRIMIR "Cantidad de repeticiones del número 10:", contador
RETORNAR
```

El programa del ejercicio esta formado por 6 subprogramas

La cantidad de elementos de la matriz es variable, pues n y m pueden tomar distintos valores para ejecuciones diferentes.

Los algoritmos de cálculo, MAXIMO, MINIMO y VECES-10, son similares a otros analizados anteriormente. Podemos apreciar en cada recorrido realizado a la matriz, como los ciclos PARA están delimitados en la cantidad de repeticiones, por los valores asignados a n y m.

Ejercicio 4

Crear una matriz de 25 filas y 5 columnas con las notas obtenidas por 25 alumnos durante el primer semestre del año.

La columna 1 se carga con el número de alumno (del 1 al 25).

Las columnas 3, 4 y 5 deben completarse con las calificaciones en las tres asignaturas cursadas: Matemática, Física y Química . La columna 2 se carga con el promedio de las tres notas.

Se pide imprimir:

a) La matriz ordenada por promedio en forma descendente.

b) La nota promedio general de Matemática, para los 25 alumnos .

```
COMIENZO
INICIO
CARGA
ORDEN
PROMEDIO
IMPRESION
FIN

INICIO:
notas(25,5)
RETORNAR

CARGA:
PARA i = 1 A 25
        notas(i,1) = i
        INGRESAR "Ingrese la nota de Matemática:", nota(i,3)
        INGRESAR "Ingrese la nota de Física:", nota(i,4)
        INGRESAR "Ingrese la nota de Química:", nota(i,5)
        nota(i,2) = ( nota(i,3) + nota(i,4) + nota(i,5) ) / 3
PROXIMO
RETORNAR

ORDEN:
mat(25,5)
cota = 25
k = 1
```

```
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                SI mat(i,2) < mat(i + 1,2) ENTONCES
                        PARA j = 1 A 5
                                aux = mat(i,j)
                                mat(i,j) = mat(i + 1,j)
                                mat(i + 1,j) = aux
                        PROXIMO
                        k = i
                FIN SI
        PROXIMO
        cota = k
REPETIR
RETORNAR

PROMEDIO:
suma = 0
PARA i = 1 A 25
        suma = suma + nota(i,3)
PROXIMO
promat = suma / 25
RETORNAR

IMPRESION:
IMPRIMIR "Matriz ordenada por promedio"
PARA i = 1 A 25
        IMPRIMIR "Número de alumno", nota(i,1)
        IMPRIMIR "Promedio:", nota(i,2)
        IMPRIMIR "Matemática:", nota(i,3)
        IMPRIMIR "Física:", nota(i,4)
        IMPRIMIR "Química:", nota(i,5)
PROXIMO
IMPRIMIR "Promedio general de Matemática:", promat
RETORNAR
```

El programa está compuesto por 5 subprogramas: INICIO, CARGA, el subprograma de ordenamiento ORDEN, un cálculo del promedio de una

columna en particular PROMEDIO y el subprograma de impresión también llamado IMPRESION.

En el subprograma de carga se utilizó un solo ciclo, pues los valores se cargaron columna a columna. Lo mismo sucedió al calcular el promedio general de Matemática.

El ordenamiento responde al algoritmo analizado en el punto anterior, con la diferencia que ahora se ordena tomando como referencia la columna 2 que contiene los promedios.

Ejercicio 5

a) Crear una matriz de 10 filas y 15 columnas. Las primeras 9 filas se cargan con valores ingresados externamente. La fila 10 debe contener la sumatoria de las 15 columnas para las 9 filas restantes.

Se pide imprimir:

a) El valor máximo de la fila 10.
b) La cantidad de ceros de la columna 15.

```
COMIENZO
INICIO
CARGA
MAXIMO
CEROS
IMPRESION
FIN

INICIO:
mat(10,15)
PARA j = 1 A 15
        mat(10,j) = 0
PROXIMO
RETORNAR

CARGA:
PARA i = 1 A 9
        PARA j = 1 A 15
                INGRESAR "Ingrese un número:", mat(i,j)
```

```
                mat(10,j) = mat(10,j) + mat(i,j)
        PROXMO
PROXIMO
RETORNAR

MAXIMO:
max = mat(10,1)
PARA j = 2 A 15
        SI mat(10,j) > max ENTONCES max = mat(10,j)
PROXIMO
RETORNAR

CEROS:
contador = 0
PARA i = 1 A 10
        SI mat(i,15) = 0 ENTONCES contador = contador + 1
PROXIMO
RETORNAR

IMPRESION:
IMPRIMIR "El valor máximo de la fila 10 es:", max
IMPRIMIR " La cantidad de ceros de la columna 15 es:", contador
RETORNAR
```

El programa está dividido en 5 subprogramas: INICIO, CARGA, MAXIMO, CEROS e IMPRESIÓN.

Presenta la particularidad que la fila 10 se obtiene como sumatoria de las 9 filas restantes. Podríamos decir que la fila 10 no es otra cosa que un conjunto de 15 acumuladores. Lo que se acumula son los 9 elementos que conforman cada una de las 9 primeras filas.

De la misma manera, tenemos el contador que cuenta la cantidad de ceros de la columna 15. Vemos como para recorrerla tenemos que fijar el índice j en 15.

RESUMEN

Una matriz es un conjunto de datos homogéneos organizados bajo un mismo nombre, cada uno de los cuales debe referenciarse por dos índices, **i** y **j**, también llamados subíndices, que identifican las filas y las columnas de la matriz **i x j**.

Ambos índices deben ser enteros mayores o iguales a cero.

Para dimensionar una matriz se necesitan los dos índices:

matriz (i , j) donde:

- i , es la cantidad de filas.
- j , la cantidad de columnas.
- matriz, el nombre de la matriz mediante el cual la computadora va a reconocer a ese conjunto de datos. Debe cumplir la misma normativa que una variable.

Los elementos de una matriz están organizados en filas y columnas como si fueran un grupo de j vectores de i elementos cada uno o viceversa.

Los elementos de una fila tienen todos ellos igual valor para el primer índice, i mientras que los de una columna tienen igual valor para el segundo, j.

El hecho de asignar el primer índice a las filas y el segundo a las columnas, es sólo una cuestión de nomenclatura. No hay ningún impedimento en asignar i a las columnas y j a las filas, siempre y cuando seamos coherentes en todo el programa.

El tipo de elementos que se almacenan en las matrices es el mismo en todas sus posiciones, todos numéricos o todos alfanuméricos, pro nunca una mezcla de ambos.

- **Ingreso de datos**

```
PARA i = 1 A f
        PARA j = 1 A c
                INGRESAR " Ingresar dato:", mat(i,j)
        PROXIMO j
```

```
PROXIMO i
```

- **Inicialización**

```
PARA i = 1 A f
        PARA j = 1 A c
                        mat(i,j) = 0
        PROXIMO j
PROXIMO i
```

- **Lectura para un cálculo**

```
PARA i = 1 A f
        PARA j = 1 A c
                        suma = suma + mat(i,j)
        PROXIMO j
PROXIMO i
```

- **Lectura para impresión**

```
PARA i = 1 A f
        PARA j = 1 A c
                        IMPRIMIR" Resultado:", mat(i,j)
        PROXIMO j
PROXIMO i
```

- **Búsqueda de un máximo**

```
max = mat(1,1)
PARA i = 1 A f
        PARA j = 1 A c
                        SI mat(i,j) > max ENTONCES max = mat(i,j)
        PROXIMO j
PROXIMO i
```

- **Búsqueda de un mínimo**

```
min = mat(1,1)
PARA i = 1 A f
```



```
        PARA j = 1 A c
                        SI mat(i,j) > min ENTONCES min = mat(i,j)
        PROXIMO j
PROXIMO i
```

- **Ordenamiento**

```
mat(n,m)
cota = n
k = 1
HACER MIENTRAS k <> 0
        k = 0
        PARA i = 1 A cota -1
                        SI mat(i,1) > mat(i + 1,1) ENTONCES
                                PARA j =1 A m
                                        aux = mat(i,j)
                                        mat(i,j) = mat(i + 1,j)
                                        mat(i + 1,j) = aux
                                PROXIMO
                                k = i
                        FIN SI
        PROXIMO
        cota = k
REPETIR
```





Ejercicios Propuestos

1. Crear una matriz mat de n filas y m columnas, cuyos elementos sean múltiplos de
Se pide imprimir:

a) La matriz completa.
b) Los elementos de la fila del medio.

2. Crear una matriz de mat de 7 filas y 9 columnas.
Imprimir:

a) La suma de las columnas pares.
b) La suma de las filas impares.

3. Crear una matriz mat1 de 8 filas y 12 columnas en la que los elementos de las filas pares sean múltiplos de 2 y los de las filas impares sean 5 por la posición de la columna.
A partir de ella crear otra matriz mat2 cuyas columnas impares sean las de mat1 elevadas al cuadrado y las pares las de mat1.

4. Construir una matriz ma de 5 filas y 5 columnas de tal forma que los elementos de la diagonal sean unos y los demas elementos se ingresen desde afuera. Imprimirla.

5. Construir una matriz mat de 15 filas y 15 columnas de tal forma que los elementos de la diagonal sean unos y los demás elementos sean ceros. Imprimirla.

6. Construir una matriz de 10 filas y 9 columnas, de tal forma que el valor de los elementos de cada columna sean, respectivamente, las tablas de multiplicar del 1 al 9.

7. Cargar una matriz de 12 filas y 31 columnas donde las filas represen-

ten los meses del año y las columnas la temperatura media de cada día.
Se pide calcular e imprimir:

a) La temperatura media de agosto.
b) El día más caluroso del año.
c) El día más frío del año.
d) El día más frío de julio.
c) El día más caluroso de enero.

Ejercicios Combinados



8.1 INTRODUCCIÓN

A lo largo del texto se han analizado los distintos tipos de estructuras, ya sea desde el punto de vista de los tipos de instrucciones, como desde el punto de vista de organización de los datos.

Como corolario del desarrollo de este curso de introducción a la programación, a partir de algoritmos escritos en pseudocódigo, y de la construcción y manipulación de estructuras internas como los vectores y las matrices; vamos a desarrollar en este capítulo un conjunto de ejercicios que combinan el manejo de vectores y matrices.

Lógicamente que esto nos obliga a integrar los conceptos estudiados en todas las unidades temáticas vistas hasta ahora, lo que desde la perspectiva del aprendizaje significativo, a alcanzar por el estudiante, es altamente positivo.

8.2 EJERCICIOS RESUELTOS

8.2.1 Ejercicios con carga con ciclo PARA...PROXIMO

Ejercicio 1

Una empresa de productos alimenticios posee 30 empleados. Los empleados concurren diariamente a trabajar en el horario de 9 a 18 hs. Mensualmente cobran un sueldo que depende de las horas trabajadas.

La empresa tiene un registro de los sueldos abonados a cada empleado en cada mes del año, en planillas mensuales, de la siguiente forma:

MES:

NÚMERO DEL EMPLEADO	CANTIDAD DE HORAS	VALOR DE LA HORA

Se tienen cargados en un vector llamado nombre$, los nombres de los 30 empleados.

Se pide calcular e imprimir:

a) Sueldo anual de cada empleado.
b) Total de sueldos pagados cada mes.
c) Máximo sueldo pagado en cada mes.
d) Porcentaje que representa cada sueldo anual sobre el total
e) Nombre del empleado que haya cobrado el mínimo sueldo en el primer semestre del año.
f) Cantidad de meses en los que se haya pagado menos de 45.000$ en total, en concepto de sueldos.

Se pide además ordenar los sueldos anuales en forma ascendente e imprimirlos de la siguiente forma:

```
NOMBRE      SUELDO ANUAL

COMIENZO
INICIO
CARGA
SUELDO-ANUAL
SUELDO-MENSUAL
PORCENTAJE
MINIMO-SEMESTRAL
MENOS-45000
ORDEN
IMPRESION
FIN
```

El programa se dividió en 9 subprogramas.

```
INICIO:
sdo(12,30)
sae(30) : nom$(30) : por(30) : nomord$(30)
sm(12) : max(12)
PARA j = 1 A 30
   INGRESAR "Nombre:", nom$(j)
   nomord$(j) = nom$(j)
```



```
PROXIMO
RETORNAR
```

Se utilizará una matriz sueldos sdo(12,30), de 12 filas para los meses del año y 30 columnas para cada uno de los empleados. En ella se cargarán los sueldos que figuran en la planilla.

Se trabajará con 6 vectores, 4 de 30 elementos y 2 de 12 elementos:

sae(30): sueldo anual de cada empleado.
nom$(30): nombre de cada empleado.
nom$(30): nombre de cada empleado, de utilización exclusiva en el ordenamiento.
sm(12): total de sueldos pagados cada mes.
max(12): máximo sueldo pagado en cada mes.
por(30): porcentaje que representa cada sueldo anual sobre el total.

```
CARGA:
PARA i = 1 A 12
   INGRESAR "Mes:", mes
   PARA j = 1 A 30
         INGRESAR "Número de empleado:", nro
         INGRESAR "Cantidad de horas:", horas
         INGRESAR "Valor de la hora:", vh
         sdo(mes,nro) = horas * vh
   PROXIMO j
PROXIMO i
RETORNAR
```

Luego del primer ciclo PARA, se ingresa el mes para no depender la carga del orden de las planillas. La matriz sueldo se carga con el producto entre las horas trabajadas y el valor de la hora.

```
SUELDO-ANUAL:
PARA j = 1 A 30
   PARA i = 1 A 12
         sae(j) = sae(j) + sdo(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

El vector sae(j) acumula la suma de todos los sueldos percibidos por el empleado durante los 12 meses. Al ejecutarse primero el ciclo j, estamos recorriendo la matriz por columna o sea por empleado.

```
SUELDO-MENSUAL:
PARA i = 1 A 12
   PARA j = 1 A 30
        sm(i) = sm(i) + sdo(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

El vector sm(i) acumula los sueldos por mes por los 30 empleados. En este caso la matriz se recorre por filas.

```
MAXIMO-MENSUAL:
PARA i = 1 A 12
   max(i) = sdo(i,1)
   PARA j = 2 A 30
        SI sdo(i,j) > max(i) ENTONCES max(i) = sdo(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Debemos calcular 12 máximos, por eso el primer ciclo PARA indica esta repetición del cálculo. Luego para cada mes inicializamos el vector de máximo con el sueldo del primer empleado: max(i) = sdo(i,1), y comparamos con los 29 restantes arrancando el segundo ciclo de 2: PARA j = 2 A 30.

```
PORCENTAJE:
sdotot = 0
PARA j = 1 A 30
   sdotot = sdotot + sae(j)
PROXIMO
PARA j = 1 A 30
   por(j) = sae(j) / sdotot * 100
PROXIMO
RETORNAR
```


Los 30 porcentajes se obtienen dividiendo el sueldo anual de cada empleado sae(j) por el total pagado en el año a todos los empleados. Esta suma se acumula en la variable sdotot.

```
MINIMO-SEMESTRAL:
minsem = sdo(1,1)
PARA i = 1 A 6
   PARA j = 1 A 30
        SI sdo(i,j) < minsem ENTONCES minsem = sdo(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

El mínimo semestral se determinó recorriendo la matriz hasta a fila 6 (enero a junio).

```
MENOS-45000:
cm = 0
PARA i = 1 A 12
   SI sm(i) < 45000 ENTONCES cm = cm + 1
PROXIMO
RETORNAR
```

El contador cm, lleva la cuenta de la cantidad de meses que se pagó menos de 45000$, en concepto de sueldos.

```
ORDEN:

cota = 30
k = 1
HACER MIENTRAS k <> 0
   k = 0
   PARA j = 1 A cota - 1
        SI sae(j) > sae(j+1) ENTONCES
             aux = sae(j)
             sae(j) = sae(j+1)
             sae(j+1) = aux
             aux$ = nomord$(j)
```

```
            nomord$(j) = nomord$(j+1)
            nomord$(j+1) = aux$
            k = j
        FIN SI
    PROXIMO
    cota = k
REPETIR
RETORNAR
```

Se ordenaron los 30 sueldos anuales del vector sae(j) y el vector de nombres nomord$(j) asociado a aquél. El vector nomord$(j) se utiliza para el ordenamiento para evitar que un reacomodamiento en el vector nom$(j) al ser ordenado, impida calcular el punto e, donde los nombres deben estar vinculados a al cálculo del mínimo semestral.

```
IMPRESION:
IMPRIMIR "Sueldo anual de cada empleado"
PARA j = 1 A 30
    IMPRIMIR "Nombre:", nomord$(j), "sueldo:", sae(j),"$"
PROXIMO
IMPRIMIR" Total de sueldos pagados cada mes"
PARA i = 1 A 12
    IMPRIMIR "mes:", i, sm(i),"$"
PROXIMO
IMPRIMIR "Máximo sueldo pagado en cada mes"
PARA i = 1 A 12
    IMPRIMIR "mes:", i,  max(i), "$"
PROXIMO
IMPRIMIR "Porcentaje que representa cada sueldo anual sobre el total'
PARA j = 1 A 30
    IMPRIMIR "Empleado:', j,"Porcentaje:", por(j)
PROXIMO
IMPRIMIR "Nombre del empleado con menor sueldo en el primer semestre"
PARA i = 1 A 6
    PARA j =1 A 30
        SI minsem = sdo(i,j) ENTONCES IMPRIMIR nom$(j)
    PROXIMO
PROXIMO i
```



```
IMPRIMIR "Cantidad de meses en los que se pagó menos de 45000:", cm
RETORNAR
```

La impresión de todos los vectores se hizo colocando un encabezamiento antes del ciclo PARA respectivo.

Ejercicio 2

Una consultora de empresas se dedica a la comercialización de software. Tiene 15 analistas que trabajan todo el día en la empresa. En este momento la empresa tiene en marcha 10 proyectos para diferentes empresas.

A fin de mes cada analista entrega una planilla en la que consigna cuantas horas trabajó en cada proyecto.

La planilla tiene el siguiente formato:

NÚMERO DE ANALISTA:

NÚMERO DE PROYECTO	CANTIDAD DE HORAS

Cada analista trabaja en todos los proyectos obligatoriamente. Los nombres de los analistas se encuentran cargados en un vector nombre$ y el valor por hora de cada analista se deberá cargar en memoria.

Se pide calcular e imprimir:

a) Cantidad de horas mensuales de cada analista.
b) Total de horas trabajadas en cada proyecto.
c) Mínima cantidad de horas trabajadas por cada analista.
d) Sueldo mensual de cada analista.
e) Nombre del analista que haya cobrado el máximo sueldo.
f) Cantidad de analistas que hayan trabajado menos de 5 horas en algu-

no de los proyectos.

Se pide además ordenar los sueldos mensuales de cada analista en forma ascendente e imprimirla de la siguiente manera:

```
NOMBRE              SUELDO MENSUAL
```

```
COMIENZO
INICIO
CARGA
HORAS-ANALISTA
HORAS-PROYECTO
MINIMO
SUELDO
MENOS-5-HORAS
ORDEN
IMPRESION
FIN
```

```
INICIO:
horas(15,10)
nom$(15) : vh(15) : ha(15) : min(15) : sdo(15)
hp(10)
PARA i = 1 A 15
   INGRESAR "Nombre:", nom$(i)
   INGRESAR" Valor de la hora:", vh(i)
PROXIMO
RETORNAR
```

El programa está dividido en 9 subprogramas. Se trabajará con una matriz horas de 15 filas una para cada analista, y 10 columnas una para cada proyecto.

Se dimensionaron 6 vectores:

nom$(15): nombre de los analistas.
vh(15): valor de la hora de cada analista.
ha(15): horas mensuales trabajadas por analista.
min(15): mínima cantidad de horas trabajadas por cada analista.
sdo(15): sueldo mensual de cada analista.

hp(10): horas trabajadas en cada proyecto.

Los valores de las horas y el nombre de los analistas se cargan desde el teclado.

```
CARGA:
PARA i = 1 A 15
   INGRESAR "Número de analista:", na
   PARA j = 1 A 10
         INGRESAR "Número de proyecto:", np
         INGRESAR "Cantidad de horas:", ch
         horas(na,np) = ch
   PROXIMO j
PROXIMO i
RETORNAR
```

Al igual que en el ejercicio anterior, el ingreso del número de analista se hace luego del primer ciclo PARA, para independizar las planillas de cualquier secuencia de carga.

```
HORAS-ANALISTA:
PARA i = 1 A 15
   PARA j = 1 A 10
      ha(i) = ha(i) + horas(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Se recorrió la matriz por filas para acumular las horas trabajadas por analista.

```
HORAS-PROYECTO:
PARA j = 1 A 10
   PARA i = 1 A 15
      hp(j) = hp(j) + horas(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

En este caso se recorre la matriz por columnas para acumular las horas por proyecto.

```
MINIMO:
PARA i = 1 A 15
    min = horas(i,1)
    PARA j = 2 A 10
        SI horas(i,j) < min ENTONCES min = horas(i,j)
    PROXIMO j
PROXIMO i
RETORNAR
```

El mínimo se inicia con el valor de las horas del primer proyecto: horas(i,1), para comparar luego con los 9 restantes: PARA j = 2 A 10.

```
SUELDO:
PARA i = 1 A 15
    sdo(i) = ha(i) * vh(i)
PROXIMO
RETORNAR
```

El sueldo de cada analista se obtiene directamente multiplicando las horas trabajadas por el valor de la hora.

```
MENOS-5-HORAS:
contador = 0
PARA i = 1 A 15
k = 0
    PARA j = 1 A 10
        SI horas(i,j) < 5 ENTONCES k = 1
    PROXIMO j
    SI k = 1 ENTONCES contador = contador + 1
PROXIMO i
RETORNAR
```

La variable contador, cuenta la cantidad de analistas que hayan trabajado menos de 5 horas en alguno de los proyectos.

Puede suceder que alguno de los analistas cumpla con esta posibilidad más de una vez; por lo cual deberíamos contarlo sólo una. La solución es

utilizar una variable switch k, como la utilizada en el algoritmo de ordenamiento.

El ciclo PARA j = 1 A 10, busca entre los 10 proyectos, si alguno cumple con la condición de menos de 5 horas, cuando encuentra un caso coloca el switch a 1.

Recorridos los 10 proyectos, el contador se incrementará una sola vez por analista cuando encuentre que k es igual a 1.

```
ORDEN:
cota = 15
k = 1
HACER MIENTRAS k <> 0
  k = 0
  PARA i = 1 A cota - 1
        SI sdo(i) > sdo(i+1) ENTONCES
              aux = sdo (i)
              sdo (i) = sdo (i+1)
              sdo (i+1) = aux
              aux$ = nom$(i)
              nom$(i) = nom$(i+1)
              nom$(i+1) = aux$
              k = i
        FIN SI
  PROXIMO
  cota = k
REPETIR
RETORNAR
```

Se ordenó el vector sueldos sdo() y se intercambió el vector nombres nom$() asociado al primero.

El ordenamiento permite en este caso responder también al punto e, pues el sueldo máximo será, el último elemento del vector ordenado.

```
IMPRESION:
PARA i = 1 A 15
   IMPRIMIR "Número de analista:", i
   IMPRIMIR" Horas trabajadas:", ha(i)
   IMPRIMIR" Mínimo de horas trabajadas:", min(i)
PROXIMO
```

```
PARA j = 1 A 10
   IMPRIMIR" Número de proyecto:", j
   IMPRIMIR" Horas trabajadas:", hp(j)
PROXIMO
IMPRIMIR "Sueldos mensuales:"
PARA i = 1 A 15
   IMPRIMIR" Nombre:", nom$(i), "Sueldo:", sdo(i), "$"
PROXIMO
IMPRIMIR "Analistas con sueldo máximo"
PARA i = 1 A 15
   SI sdo(i) = sdo(15) ENTONCES IMPRIMIR nom$(i)
PROXIMO
RETORNAR
```

8.2.2 Ejercicios con carga con ciclo HACER MIENTRAS...REPETIR

Ejercicio 3

Una empresa se dedica a la comercialización de combustibles. Cuenta con 15 estaciones de servicio distribuidas en Capital y Gran Buenos Aires. (Los nombres de las estaciones se deberán cargar en un vector en memoria).

En cada estación de servicio se venden los siguientes productos:

1 - Nafta común.
2 - Nafta especial.
3 - Nafta sin plomo.
4 - Gas Oil.

Mensualmente cada estación remite a la central, los comprobantes de las ventas en pesos con los siguientes datos datos:

- Número de estación.
- Código de combustible.
- Importe de la venta.

Se pide calcular e imprimir:

a) Total recaudado por cada estación.
b) Total recaudado por cada tipo de combustible.
c) Nombre del combustible que recaudó menos, para cada estación.
d) Máxima recaudación de cada combustible.
e) Estación de mayor recaudación.
f) Cantidad de estaciones que hayan facturado más de 15000$en total para los 4 combustibles.
g) Porcentaje de facturación y nombre del combustible de porcentaje máximo,

```
COMIENZO
INICIO
CARGA
REC-ESTACION
REC-COMBUSTIBLE
MIN-ESTACION
MAX-COMBUSTIBLE
MAX-ESTACION
FACTURACION-15000
PORCENTAJE
FIN
```

El programa se divide en 9 subprogramas.

```
INICIO:
venta(15,4)
re(15) : mc(15) : ne$(15)
rc(4) : cr$(4) : maxc(4) : por(4)
nc$(1) = "Nafta Común"
nc$(2) = "Nafta Especial"
nc$(3) = "Nafta sin Plomo"
nc$(4) = "Gas Oil"
PARA i = 1 A 15
   INGRESAR "Nombre de estación:", ne$(i)
PROXIMO
RETORNAR
```

Se creó la matriz venta, que almacenará las ventas de los 4 combustibles

para las 15 estaciones.

Se dimensionaron 7 vectores:

re(15): recaudación por estación.
mc(15): menor recaudación de un combustible por estación.
ne$(15): nombre de cada estación.
rc(4): recaudación por combustible.
cn$(4): nombre de cada combustible.
maxc(4): máximo recaudación de cada combustible.
por(4): porcentaje de facturación de cada combustible.

```
CARGA:
INGRESAR "Número de estación:", est
HACER MIENTRAS est <> 0
   INGRESAR "Código de combustible:", cod
   INGRESAR "Venta en pesos:", vta
   venta(est,cod) = vta
   INGRESAR "Número de estación:", est
REPETIR
RETORNAR
```

A diferencia de los ejemplos de carga vistos en los ejercicios anteriores, ahora desconocemos la cantidad de comprobantes remitidos por todas las estaciones. Recurrimos a utilizar una de las variables como fin de archivo; en este caso, el número de estación. Cuando no queden más comprobantes por procesar ingresaremos un cero que indicará el final de la carga de datos.

```
REC-ESTACION:
PARA i = 1 A 15
   PARA j = 1 A 4
         re(i) = rc(i) + venta(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

La recaudación por estación se calcula recorriendo la matriz por filas.

```
REC-COMBUSTIBLE:
PARA j = 1 A 4
   PARA i = 1 A 15
```



```
         rc(j) = rc(j) + venta(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

En este caso se debe recorrer por columnas para determinar la recaudación por combustible.

```
MIN-ESTACION:
PARA i = 1 A 15
   mc(i) = venta(i,1)
   PARA j = 2 A 4
         SI venta(i,j) < mc(i) ENTONCES mc(i) = venta(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Se obtuvieron 15 mínimos, uno para cada estación, referidos al combustible de menor recaudación.

Se comienza suponiendo para cada estación que la nafta común es a de menor recaudación: mc(i) = venta(i,1), y se compara su valor con las tres restantes: PARA j = 2 A 4.

```
MAX-COMBUSTIBLE:
PARA j = 1 A 4
   maxc(j) = venta(1,j)
   PARA i = 2 A 15
         SI venta(i,j) > maxc(j) ENTONCES maxc(j) = venta(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

Se hallaron 4 máximos entre las 15 estaciones, uno para cada combustible.

Se comienza asignando para cada combustible, a la estación 1 como máximo: maxc(j) = venta(1,j) y luego se compara con las 14 restantes.

```
MAX-ESTACION:
remax = re(1)
PARA i = 2 A 15
   SI re(i) > remax ENTONCES remax = re(i)
PROXIMO
RETORNAR
```

En este subprograma se calcula la máxima recaudación entre las 15 estaciones. Al ser un valor único, se almacena esta recaudación en remax, para luego localizar en el subprograma de impresión, las estaciones que responden a ese máximo.

```
FACTURACION-15000:
ce15 = 0
PARA i = 1 A 15
   SI re(i) > 15000 ENTONCES ce15 = ce15 + 1
PROXIMO
RETORNAR
```

La variable ce15 es un contador que almacena la cantidad de estaciones que facturaron más de 15000$. Se trabaja comparando el vector re(), pues contiene las recaudaciones de todas las estaciones

```
PORCENTAJE:
tf = 0
PARA j = 1 A 4
   tf = tf + rc(j)
PROXIMO
PARA j = 1 A 4
   por(j) = rc(j) / tf * 100
PROXIMO
pmax = por(1)
PARA j = 2 A 4
   SI por(j) > pmax ENTONCES pmax = por(j)
PROXIMO
RETORNAR
```

El porcentaje de facturación de cada combustible está determinado por la expresión: por(j) = rc(j) / tf * 100 ,donde tf es el acumulador de recaudaciones donde se almacenará el total recaudado por todos los combustibles y rc(j) la recaudación individual de cada uno de los 4 combustibles.


```
IMPRESION:
IMPRIMIR "Recaudación por estación"
PARA i = 1 A 15
IMPRIMIR "Estación:", i, "Recaudación:", re(i), "$"
PROXIMO
IMPRIMIR "Recaudación por combustible"
PARA j = 1 A 4
   IMPRIMIR "Combustible:", nc$(j), "Recaudación:", rc(j), "$"
PROXIMO
IMPRIMIR "Combustible de menor recaudación por estación."
PARA i = 1 A 15
   IMPRIMIR "Estación número:", i
   PARA j = 1 A 4
         SI venta(i,j) = mc(i) ENTONCES IMPRIMIR nc$(j)
   PROXIMO j
PROXIMO i
IMPRIMIR "La recaudación máxima por combustible"
PARA j = 1 A 4
   IMPRIMIR "Combustible:", nc$(j), "Recaudación maxima:", maxc(j),"$"
PROXIMO
IMPRIMIR "Estación de máxima recaudación"
PARA i = 1 A 15
   SI re(i) = remax ENTONCES IMPRIMIR ne$(i)
PROXIMO
IMPRIMIR "Estaciones con facturación mayor a 15000$:", ce15
IMPRIMIR "Porcentaje de facturación de cada combustible"
PARA j = 1 A 4
   IMPRIMIR "Combustible:", nc$(j), "Porcentaje:", por(j),"%"
PROXIMO
IMPRIMIR "Combustible de mayor porcentaje:"
PARA j = 1 A 4
   SI por(j) = pmax ENTONCES IMPRIMIR nc$(j)
PROXIMO
RETORNAR
```

8.2.3 Ejercicios con dos matrices

Ejercicio 4

Un instituto de computación vende cursos a través de 15 vendedores. Los vendedores salen diariamente a vender los cursos en las empresas.

Los cursos que se ofrecen son 10 y sus nombres se deberán cargar en memoria al igual que los nombres de los vendedores y los precios de los cursos.

Cada vendedor llena a fin de mes un formulario con la siguiente información:

NÚMERO DE VENDEDOR:

NÚMERO DE CURSO	CANTIDAD DE ALUMNOS

Se pide calcular e imprimir:

a) Total de alumnos inscriptos en cada curso.
b) Cantidad de alumnos atendidos por cada vendedor.
c) Nombre del curso que haya tenido más inscriptos.
d) Máxima cantidad de alumnos inscriptos por cada vendedor y nombre del curso al que pertenece esa cantidad.
e) Recaudación por vendedor.
f) Recaudación por curso.
g) Nombre del vendedor que haya inscripto la mayor cantidad de alumnos, en una inscripción.
h) Cantidad de cursos para los cuales no realizó inscripciones el vendedor 5.


```
COMIENZO
INICIO
CARGA
ALUMNOS-CURSO
ALUMNOS-VENDEDOR
CURSO-MAS-INSCRIPTOS
MAX-VENDEDOR
REC-VENDEDOR
REC-CURSO
MAX-UNA-INSCRIPCION
VENDEDOR-5
IMPRESION
FIN
```

El programa está formado por 11 subprogramas.

```
INICIO:
cantalu(15,10) : recalu(15,10)
nv$(15) : av(15) : maxv(15) : rv(15)
nc$(10) : ac(10) : rc(10): pc(10)
PARA i = 1 A 15
   INGRESAR "Nombre del vendedor:", nv$()
PROXIMO
PARA j = 1 A 10
   INGRESAR "Nombre del curso:", nc$(j)
INGRESAR "Precio del curso:", pc(j)
PROXIMO
RETORNAR
```

Por primera vez vamos a trabajar con 2 matrices, una almacenará la cantidad de alumnos cantalu(15,10), y la otra la recaudación recalu(15,10), en ambos casos para 15 vendedores y 10 cursos.

Dispondremos de 7 vectores:

nv$(15): nombre de los 15 vendedores.
av(15): alumnos por vendedor.
maxv(15): máxima cantidad de alumnos inscriptos por vendedor.
rv(15): recaudación por vendedor.
nc$(10): nombre de los 10 cursos.

```
ac(10): alumnos por curso.
rc(10): recaudación por curso.
pc(10): precio por curso.

CARGA:
PARA i = 1 A 15
   INGRESAR "Número de vendedor:", nrov
   PARA j = 1 A 10
         INGRESAR "Número de curso:", nroc
         INGRESAR "Cantidad de alumnos:", ca
         cantalu(nrov,nroc) = ca
         recalu(nrov,nroc) = ca * pc(j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Se cargaron las dos matrices en forma paralela: la cantidad de alumnos cantalu(nrov,nroc) = ca, y la recaudación recalu(nrov,nroc) = ca * pc(j).

```
ALUMNOS-CURSO:
PARA j = 1 A 10
   PARA i = 1 A 15
         ac(j) = ac(j) + cantalu(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

Se recorrió la matriz por columnas para los 10 cursos.

```
ALUMNOS-VENDEDOR:
PARA i = 1 A 15
   PARA j = 1 A 10
         av(j) = av(j) + cantalu(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Ahora se recorrió la matriz por filas para los 15 vendedores

CURSO-MAS-INSCRIPTOS:


```
insmax = ac(1)
PARA j = 2 A 10
   SI ac(j) > insmax ENTONCES insmax = ac(j)
PROXIMO
RETORNAR
```

La variable insmax, almacena la mayor inscripción para un curso. En la impresión se determinará el nombre del curso que responde a esa inscripción.

```
MAX-VENDEDOR:
PARA i = 1 A 15
   maxv(i) = cantalu(i,1)
   PARA j = 2 A 10
         SI cantalu(i,j) > maxv(i) ENTONCES maxv(i) = cantalu(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Para cada vendedor hay que calcular la máxima cantidad de inscriptos, valor que se almacena en maxv(i). Este vector se inicializa 15 veces con la cantidad de alumnos del primer curso para luego compararlo con los 9 restantes.

```
REC-VENDEDOR:
PARA i = 1 A 15
   PARA j = 1 A 10
         rv(j) = rv(j) + recalu(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

El vector rv(j) acumula la recaudación total por vendedor, para esto se recorre la matriz por filas.

```
REC-CURSO:
PARA j = 1 A 10
   PARA i = 1 A 15
         rc(j) = ar(j) + recalu(i,j)
```

```
    PROXIMO i
  PROXIMO j
  RETORNAR
```

El vector rc(j) acumula la reccudación total por vendedor, ahora se recorre la matriz por columnas.

```
  MAX-UNA-INSCRIPCION:
  maxins = cantalu 1,1)
  PARA i = 1 A 15
    PARA j = 1 A 10
          SI cantalu(i,j) > maxins ENTONCES maxins = cantalu(i,j)
    PROXIMO j
  PROXIMO i
  RETORNAR
```

El máximo calculado en este caso es de toda la matriz, pues es para una sola inscripción.

```
  VENDEDOR-5:
  c5 = 0
  PARA j = 1 A 10
    SI cantalu(5,j) = 0 ENTONCES c5 = c5 + 1
  PROXIMO j
  RETORNAR
```

La variable c5 es un contador que cuenta la cantidad de cursos que no realizó inscripciones el vendedor 5. El valor de las filas en la matriz queda fijo para recorrer exclusivamente esa fila: cantalu(5,j).

```
  IMPRESION:
  IMPRIMIR "Alumnos por curso"
  PARA j = 1 A 10
    IMPRIMIR "Curso:", nc$(j), "Alumnos:",ac(j)
  PROXMO
  IMPRIMIR "Alumnos por vendedor"
  PARA i = 1 A 15
    IMPRIMIR "Vendedor:", nv$(j), "Alumnos:",av(j)
  PROXMO
```



```
  IMPRIMIR "Curso de máxima inscripción"
  PARA j = 1 A 10
    SI ac(j) = insmax ENTONCES IMPRIMIR nc$(j)
  PROXIMO
  IMPRIMIR "Máxima inscripción por vendedor"
  PARA i = 1 A 15
    IMPRIMIR "Vendedor:", nv$(i), "Máxima inscripción:", maxv(i)
    IMPRIMIR "Curso:"
    PARA j = 1 A 10
    SI maxv(i) = cantalu(i,j) ENTONCES IMPRIMIR nc$()
    PROXIMO j
  PROXIMO i
  IMPRIMIR "Recaudación por vendedor"
  PARA i = 1 A 15
    IMPRIMIR "Vendedor:", nv$(i), "Recaudación:",rv(i)
  PROXIMO
  IMPRIMIR "Recaudación por curso"
  PARA j = 1 A 10
    IMPRIMIR "Curso:', nc$(j), "Recaudación:",rc(j)
  PROXIMO
  IMPRIMIR "Vendedor de máxima inscripción:"
  PARA i = 1 A 15
    k = 0
    PARA j = 1 A 10
        SI maxins = cantalu(i,j) ENTONCES k = 1
    PROXIMO j
    SI k = 1 ENTONCES IMPRIMIR nv$(i)
  PROXIMO i
  IMPRIMIR "El vendedor", nv$(5), "no realizo inscripciones en:", c5, "cursos"
  RETORNAR
```

La impresión presenta todas las variantes analizadas hasta ahora para mostrar resultados. Nuevamente se recurre a la utilización de un switch, esta vez para condicionar la impresión del vendedor de máxima inscripción (para una), pues puede darse el caso de que el mismo vendedor halla logrado más de una vez la máxima inscripción y podría imprimirse dos o más veces su nombre.

Ejercicio 5

La empresa tiene 17 centros de embotellado de sus gaseosas. En cada centro trabajan en promedio 120 personas.

Diariamente se procede al embotellamiento, teniendo como mínimo que llenar 10000 envases y como máximo 15000.

Las gaseosas que se envasan son:

1- COLA.
2- LIMA LIMÓN
3- NARANJA.
4- POMELO.

El costo de embotellamiento varía para cada centro y cada gaseosa.
El costo de limpieza de cada envase, de cualquier gaseosa es de 0.10$
El costo de mano de obra es de 0.05$ por envase.

Para los centros 5, 10 y 16 hay que considerar que los costos se incrementan en un 25% debido a la ubicación alejada de dichos centros.

Al costo de cada litro, se le carga el 100% para calcular el precio por litro.

Cada centro entrega al cabo de 20 días de trabajo, una planilla con la cantidad de envases llenados en ese lapso para cada gaseosa.

NÚMERO DEL CENTRO: NOMBRE DEL CENTRO:

NÚMERO DE GASEOSA	COSTO DEL LÍQUIDO	CANTIDAD ENVASADA
	(por litro)	(en litros)

Se pide calcular e imprimir:

a) Cantidad envasada por centro.

b) Cantidad envasada de cada gaseosa.
c) Costo de embotellamiento de cada centro.
d) Para cada gaseosa calcular el nombre del centro en el que se envasaron mayor cantidad de litros.
e) Porcentaje que representa el costo de envasado de cada centro sobre el costo total de la empresa.
f) Si consideramos que se vende todo lo que se produce, ¿Cuál fue la facturación total de cada centro?
g) Nombre de la gaseosa de mayor facturación para los 17 centros.

Imprimir los items a, c y f ordenados en forma ascendente por facturación.

```
COMIENZO
INICIO
CARGA
CANTIDAD-CENTRO
CANTIDAD-GASEOSA
COSTO-CENTRO
MAXIMO-GASEOSA
PORCENTAJE
FACTURACION-CENTRO
MAXIMO-CENTRO
ORDEN
IMPRESION
FIN
```

Vamos a trabajar con 11 subprogramas.

```
INICIO:
costo(17,4) : canti(17,4)
nc$(17) : nroc(17) : cc(17) : fc(17) : maxc(17) : ccc(17)
ng$(4) : ccg(4) : maxg(4) : por(17)
PARA i = 1 A 17
   INGRESAR "Nombre del centro:", nc$(i)
   nroc(i) = i
PROXIMO
ng$(1) = "Cola"
```

```
ng$(2) = "Lima limón"
ng$(3) = "Naranja"
ng$(4) = "Pomelo"
RETORNAR
```

Se trabaja nuevamente con dos matrices, una para los costos: costo(17,4), y otra para almacenar la cantidad de litros: canti(17,4). Ambas serán de 17 filas, una por centro, y 4 columnas.

Los vectores a utilizar son 9:

nc$(17): nombre de cada centro.
nroc(17):número de centro.
cc(17):costo por centro.
fc(17):facturación por centro
maxc(17): gaseosa de máximo envasado por centro.
cec(17):cantidad envasada por centro
ng$(4):nombre de cada gaseosa.
ceg(4):cantidad envasada por gaseosa.
maxg(4): centro de mayor envasado para cada gaseosa.
por(17): porcentaje que representa el costo de envasado.

```
CARGA:
PARA i = 1 A 17
   INGRESAR "Número de centro:", cen
   PARA j = 1 A 4
         INGRESAR "Número de gaseosa:", gas
         INGRESAR "Costo del líquido:", cl
         INGRESAR "Cantidad envasada:", ce
         canti(cen,gas) = ce
         costo(cen,gas) = (cl + 0.10 + 0.05) * ce
         SI cen = 5 OR cen = 10 OR cen = 16 ENTONCES
              costo(cen,gas) = costo(cen,gas) * 1.25
   FIN SI
PROXIMO j
PROXIMO i
```



```
RETORNAR
```

Se cargaron las dos matrices; canti(i,j) directamente con la cantidad envasada: canti(cen,gas) = ce y la matriz costo(i,j) con el valor que surge de sumar los costos de el líquido por litro, limpieza del envase y mano de obra por envase, y multiplicarlos por la cantidad envasada en litros: costo(cen,gas) = (cl + 0.10 + 0.05) * ce. Además el costo se incrementa un 25%, costo(cen,gas) = costo(cen,gas) * 1.25, si los centros son el 5 o el 10 o el 16 por lo que se utiliza una OR para unir las condiciones dentro del SI:
SI cen = 5 OR cen = 10 OR cen = 16.

```
CANTIDAD-CENTRO:
PARA i = 1 A 17
   PARA j = 1 A 4
         cec(i) = cec(i) + canti(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Se recorre la matriz por filas para acumular la cantidad de litros envasados por centro.

```
CANTIDAD-GASEOSA:
PARA j= 1 A 4
   PARA i = 1 A 17
         ceg(j) = ceg(j) + canti(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

Se recorre la matriz por columnas para acumular la cantidad de litros envasados por gaseosa.

```
COSTO-CENTRO:
PARA i = 1 A 17
   PARA j = 1 A 4
         cc(i) = cc(i) + costo(i,j)
   PROXIMO j
PROXIMO i
```

```
RETCRNAR
```

Se recorre la matriz por filas para acumular el costo de envasado por centro.

```
MAXIMO-GASEOSA:
PARA j= 1 A 4
   maxg(j) = canti(1,j)
   PARA i = 2 A 17
         SI canti(i,j) > maxg(j) ENTONCES  maxg(j) =  canti(i,j)
   PROXIMO i
PROXIMO j
RETORNAR
```

Se inicializa maxg(j) con el valor envasado para esa gaseosa por el primer centro: maxg(j) = canti(1,j)y se compara con los 16 restantes: PARA i = 2 A 17 .

```
PORCENTAJE:
ctot = 0
PARA i = 1 A 17
   ctot = ctot + cc(i)
PROXIMO
PARA i = 1 A 17
   Por(i) = cc(i) / ctot * 100
PROXIMO
RETORNAR
```

Se calcula en este caso el porcentaje que representa el costo de envasado de cada centro, cc( ), sobre el costo total de la empresa, acumulador ctot.

```
FACTURACION-CENTRO:
PARA i = 1 A 17
   PARA j = 1 A 4
         fc(i) = cc(i) * 2
   PROXIMO j
PROXIMO i
RETORNAR
```


La facturación por centro se obtiene duplicando el costo por centro: fc(i) = cc(i) * 2 , ya que sabemos que el precio final por litro surge de recargar un 100% el costo.

```
MAXIMO-CENTRO:
PARA i = 1 A 17
   maxc(i) = 2 * costo(i,1)
   PARA j = 2 A 4
         SI 2 * costo(i,j) > maxc(i) ENTONCES maxc(i) = 2 * costo(i,j)
   PROXIMO j
PROXIMO i
RETORNAR
```

Se determinó la máxima facturación de una las 4 gaseosas para los 17 centros. Se supone que el máximo corresponde a la primer gaseosa maxc(i) = 2 * costo(i,1), y se compara con las 3 restantes: PARA j = 2 A 4. Se podría haber utilizado una tercera matriz facturación, pero dado que se la utilizaría solamente en este subprograma, se prefirió trabajar con el producto por 2 de la matriz costo(i,j).

```
ORDEN:
cota = 17
k = 1
HACER MIENTRAS k <> 0
   k = 0
   PARA i = 1 A cota - 1
         SI fc(i) > fc(i+1) ENTONCES
               aux = fc(i)
               fc(i) = fc(i +1)
               fc(i +1) = aux
               aux = nroc(i)
               nroc (i) = nroc (i +1)
               nroc (i + 1) = aux
               aux = cec(i)
               cec (i) = cec (i +1)
               cec (i +1) = aux
               aux = cc(i)
               cc(i) = cc(i +1)
               cc(i +1) = aux
```

```
            k = i
        FIN SI
    PROXIMO
    cota = k
REPETIR
RETORNAR
```

Se ordenaron 4 vectores en función de la facturación de cada centro fc(i).

```
IMPRESION:
PARA i = 1 A 17
    IMPRIMIR "Centro:", noc(i), "Cantidad envasada:", cec(i), "litros",
"Costo de envasado:", cc(i), "$", "Facturación:", fc(i), "$"
PROXIMO
IMPRIMIR "Cantidad envasada por gaseosa"
PARA j = 1 A 4
    IMPRIMIR "Gaseosa:", ng$(j), ceg(j), "litros"
PROXIMO
IMPRIMIR" Centro de máxima cantidad de litros de gaseosa"
PARA j = 1 A 4
    IMPRIMIR "Gaseosa:", ng$(j)
    PARA i = 1 A 17
        SI cant(i,j) = maxg(j) ENTONCES IMPRIMIR nc$(i)
    PROXIMO i
PROXIMO j
IMPRIMIR "Porcentaje del costo de envasado de cada centro sobre el to-
tal"
PARA i = 1 A 17
    IMPRIMIR "Centro:", nc$(i)
    IMPRIMIR "Porcentaje:", por(i), "%'
PROXIMO
IMPRIMIR "Gaseosa de mayor facturación por centro"
PARA i = 1 A 17
    IMPRIMIR "Centro:", nc$(i)
    PARA j = 1 A 4
        SI 2 * costo(i,j) = maxc(i) ENTONCES IMPRIMIR ng$(i)
    PROXIMO j
PROXIMO i
RETORNAR
```




Ejercicios Propuestos

1. Una empresa se dedica a almacenamiento y posterior distribución de cereales en el interior del país. Para ello cuenta con 20 depósitos en Capital, que en general se ubican en las inmediaciones de las estaciones de ferrocarril.

(Los nombres de los depósitos se deberán cargar en un vector en memoria).

En cada depósito se pueden almacenar 4 cereales:

1 Maíz.
2 Trigo.
3 Cebada.
4 Centeno.

Mensualmente la oficina central, recibe una planilla de cada depósito con los siguientes datos:

- Número de depósito.
- Existencias de maíz.
- Existencias de trigo.
- Existencias de cebada.
- Existencias de centeno.

Se pide calcular e imprimir:

a) Cantidad total de kilos almacenados de cada cereal.
b) Cantidad de kilos almacenados en cada depósito.
c) Nombre del cereal que almacenó menor cantidad de kilos, para cada depósito.
d) Máxima cantidad de kilos almacenados de cada cereal.
e) Depósito de mayor recaudación.
f) Cantidad de depósitos que hayan almacenado más de 50000 kilos, para los 4 cereales.

g) Porcentaje de kilos de cada cereal sobre el total de kilos almacenados y nombre del cereal de porcentaje máximo.

2. Una facultad privada contrata profesores para cubrir el dictado de sus materias. Los profesores trabajan de marzo a diciembre, dependiendo su sueldo de la cantidad d horas trabajadas.

En enero y febrero, cobran un valor que consiste en la mitad del sueldo de diciembre.

La facultad lleva un registro de cada profesor, en una planilla personal con los siguientes datos:

NÚMERO DE PROFESOR:

NOMBRE:

MES	CANTIDAD DE HORAS	NÚMERO DE CATEGORÍA

Los valores de las horas dependen de la categoría y son los siguientes:

1. Profesor Titular 15$
2. Profesor Asociado 12$
3. Profesor Adjunto 10$
4. Jefe de Trabajos Prácticos 8$
5. Ayudante de 1era 6$

Se pide calcular e imprimir:

a) Sueldo anual de cada profesor.
b) Total de sueldos pagados cada mes.
c) Para cada profesor la cantidad de sueldos del año mayores al sueldo promedio de cada uno.
d) Máximo sueldo pagado en cada mes.
e) Porcentaje que representa cada sueldo anual sobre el total.

f) Nombre del profesor que haya trabajado la mayor cantidad de horas en diciembre.

Los items a, e se deben imprimir ordenados en forma ascendente de la siguiente manera:

NÚMERO DE PROFESOR	NOMBRE	SUELDO ANUAL	PORCENTAJE

3. Un laboratorio de productos medicinales, elabora 15 remedios en base a 4 drogas. El valor de cada remedio depende de la droga que utiliza. Por cada remedio, el laboratorio tiene un registro para los 12 meses del año, con los siguientes datos:

NÚMERO DE REMEDIO:
NOMBRE:
DROGA UTILIZADA: (número)
CANTIDAD: (en miligramos)
MES (número) CANTIDAD VENDIDA

Los valores de las drogas son los siguientes:

Droga 1 5$ / mg
Droga 2 2$ / mg
Droga 3 3$ / mg
Droga 4 1$ / mg

El valor del remedio depende del valor de la droga usada y de la cantidad que se utiliza de la misma. Se deberá tener en cuenta que en los meses de marzo y de septiembre, el precio de todos los remedios, sufren un descuento del 20%.

Se pide calcular e imprimir:

a) La recaudación anual de cada remedio.
b) La recaudación total del mes.
c) Para cada remedio, la cantidad de meses del año en que la recaudación superó a la recaudación promedio de todo el año.
d) La máxima recaudación de cada mes.
e) El porcentaje que representa la recaudación de cada remedio en cada mes del año sobre la recaudación anual.
f) El nombre del remedio que haya tenido menor recaudación en el segundo semestre del año.
g) El nombre del remedio que haya vendido la mayor cantidad de unidades en marzo.

Se pide además ordenar la recaudación anual en forma ascendente e imprimirla de la siguiente manera:

NRO. DEL REMEDIO	NOMBRE	REC. ANUAL	REC. PROMEDIO

4. Una empresa consultora contrata profesionales del área de sistemas para cubrir la demanda de mercado de sus clientes.

Los profesionales trabajan durante todo el año. Cada profesional cobra un sueldo básico, más la cantidad de horas trabajadas.

La empresa lleva un registro de cada profesional, en una planilla personal, con los siguientes datos:

NÚMERO DE LEGAJO:
NOMBRE:

MES	CANTIDAD DE HORAS	NÚMERO DE CATEGORÍA

Los valores del sueldo básico y de las horas dependen de la categoría y son los siguientes:

	Sueldo básico	valor por hora
1.Analista Senior	500$	15$
2.Analista Junior	400$	12$
3.Programador Senior	250$	8$
4.Programador Junior	200$	7$

Se pide calcular e imprimir:

a) Sueldo anual de cada profesional.
b) Total de sueldos pagados cada mes.
c) Para cada profesional, la cantidad de sueldos del año mayores al sueldo promedio de cada uno.
d) Máximo sueldo pagado en cada mes.
e) Porcentaje que representa cada sueldo anual sobre el total.
f) Nombre del profesional que haya cobrado el mínimo sueldo en el segundo semestre del año.
g) Nombre del profesional que haya trabajado la mayor cantidad de horas en diciembre.

Se pide además ordenar los sueldos anuales en forma ascendente e imprimirlo de la siguiente manera:

NRO DEL PROF.	NOMBRE	SUELDO ANUAL	SUELDO PROMEDIO

5. Una empresa de productos alimenticios posee 6 empleados. Los empleados concurren diariamente a trabajar en el horario de 9 a 18 hs. Mensualmente cobran un sueldo que depende de las horas trabajadas.

La empresa tiene un registro de los sueldos abonados a cada empleado en cada mes del año, en planillas mensuales, de la siguiente forma:

MES:
NÚMERO DE EMPLEADO:
CANTIDAD DE HORAS TRABAJADAS:
VALOR DE LA HORA:

En los meses de junio y diciembre, al sueldo se le deberá sumar la mitad del sueldo del correspondiente mes en concepto de aguinaldo.

Se pide calcular e imprimir:

a) Sueldo anual de cada empleado.
b) Total de sueldos pagados cada mes.
c) Para cada empleado, la cantidad de sueldos del año mayores al sueldo promedio de cada uno.
d) Máximo sueldo pagado cada mes.
e) Porcentaje que representa cada sueldo anual sobre el total.
f) Nombre del empleado que haya cobrado el mínimo sueldo en el primer semestre del año.
g) Nombre del empleado que haya trabajado la mayor cantidad de horas en diciembre.

Se pide además ordenar los sueldos anuales en forma ascendente e imprimirlo de la siguiente manera:

NRO DE EMPLEADO	NOMBRE	SUELDO ANUAL	SUELDO PROMEDIO

6. En un taller de computación se arman computadoras que se distribuyen en 5 puntos del interior del país. Los modelos que se arman son 4, cuyos costos son:

1. 300$
2. 500$
3. 700$
4. 900$

el coste de armado es de 35$ por máquina.
El coste del flete para distribución depende del lugar al que se manda:

Córdoba :	25$ por máquina.
Santa Fé:	23$ por máquina.
Chubut:	30$ por máquina.
Salta:	40$ por máquina.
Misiones:	45$ por máquina.

El taller tiene 4 planillas en las que anota a lo largo del mes, la cantidad y zona de distribución donde son enviadas las máquinas.
Dependiendo de la zona y del modelo de la máquina existen porcentajes de descuento que se anotan en la planilla a fin de mes.
El precio de cada máquina se calcula cargándole al costo total el 200%.
Cada planilla al finalizar el mes, tiene el siguiente formato:

MODELO DE MÁQUINA: (número)		NOMBRE DEL MODELO
ZONA DE DISTRIBUCIÓN	CANTIDAD	PORCENTAJE DE DESCUENTO

Se pide calcular e imprimir:

a) La cantidad vendida en cada zona.
b) La cantidad vendida de cada modelo.
c) Total facturado de cada modelo.
d) Para cada zona, el nombre del modelo del que se vendieron mayor cantidad de unidades.
e) Porcentaje que representa la facturación de cada zona sobre la facturación total de la empresa.
f) Si consideramos que se vende todo lo que se produce, cuanto se facturó a cada zona.
g) Nombre de la mayor facturación para los 4 modelos.

Los items b y c deben imprimirse ordenados de manera ascendente, por facturación de la siguiente forma:

NÚMERO DE MODELO	CANTIDAD VENDIDA	TOTAL FACTURADO

7. Una empresa de cemento, tiene 12 centros de embolsado de sus materiales. En cada centro trabajan un promedio de 120 personas.

Diariamente se procede al embolsado, teniendo como mínimo que llenar 1000 bolsas y como máximo 2000.

Los materiales que se embolsan son:

1. Cal.
2. Arena.
3. Cemento.
4. Granito.

El costo de embolsado varía para cada centro y cada material.
El costo de la bolsa de cada material es de 0.25$.
El costo de la mano de obra es de 0.15$ por bolsa.

Para los centros 5, 6 y 12, hay que considerar que los costos se incrementan en un 25% debido a la ubicación alejada de dichos centros.

Al costo de la bolsa, se le carga el 200% para calcular el precio de la bolsa.

Cada centro entrega al cabo de 20 días de trabajo, una planilla con la cantidad de bolsas llenadas en ese lapso para cada material.

NÚMERO DE CENTRO:		NOMBRE DEL CENTRO:
NRO DE MATERIAL	COSTO DE EMBOLSADO	CANTIDAD ENVASADA

Se pide:

a) Cantidad embolsada por centro.

b) Cantidad embolsada de cada materia.
c) Costo de embolsado de cada centro.
d) Para cada material, nombre del centro en el que se embolsaron mayor cantidad de bolsas.
e) Porcentaje que representa el costo de embolsado de cada centro sobre el costo total de la empresa.
f) Si consideramos que se vende todo lo que se produce, ¿ Cuál fue la facturación total de cada centro?
g) Nombre del material de mayor facturación para los 12 centros.

Se pide además imprimir los items a, c y f, ordenados de manera ascendente por facturación.

Apéndice

Resumen de Instrucciones

INSTRUCCIONES DE ENTRADA

Sintaxis INGRESAR variable

Sintaxis: INGRESAR variable1,variable2,...,variableN

Sintaxis: INGRESAR"comentario",variable

INSTRUCCIONES DE SALIDA

Sintaxis: IMPRIMIR "comentario"

Sintaxis: IMPRIMIR variable

Sintaxis: IMPRIMIR variable1,variable2,...,variableN

Sintaxis: IMPRIMIR constante

Sintaxis: IMPRIMIR expresión

Sintaxis: IMPRIMIR"comentario",variable

INSTRUCCIONES DE ASIGNACIÓN

Sintaxis: variable = constante

Sintaxis: variable = expresión

Sintaxis: variable1 = variable2

INSTRUCCIONES DE DECISIÓN

- Instrucción de decisión simple:SI......ENTONCES......

Sintaxis: SI condición ENTONCES instrucción

- Instrucción de decisión doble: SI......ENTONCES.......SINO.......

Sintaxis: SI condición ENTONCES instrucción1 SINO instrucción2

- Instrucción de decisión en bloques:

```
Sintaxis:    SI condición ENTONCES
                 instrucción(es)
             SINO
                 instrucción(es)
             FIN SI
```

SINO es opcional por lo que la instrucción puede escribirse:

```
Sintaxis:    SI condición ENTONCES
                 instrucción(es)
             FIN SI
```

- Instrucción de decisión múltiple: SELECCIONAR CASO........


```
Sintaxis:    SELECCIONAR CASO variable
                 CASO condición1
                     instrucción(es)
                 CASO condición2
                     instrucción(es)
                 CASO condición3
                     instrucción(es)
                     ................
                     ................
                 OTRO CASO
                     instrucción(es)
             FIN SELECCIONAR
```

INSTRUCCIONES DE REPETICIÓN

- Instrucción HACER MIENTRAS

```
Sintaxis:    HACER MENTRAS condición
                 instruccion(es)
             REPETIR
```

- Instrucción HACER HASTA

```
Sintaxis:    HACER
                 instruccion(es)
             REPETIR HASTA condición
```

- Instrucciones PARA...PRÓXIMO

Sintaxis: PARA variable índice = valor inicial A valor final PASO incremento

```
                 instruccion(es)
             PROXIMO
```

Bibliografía

Alcalde, Eduardo y García, Miguel: "Metodología de la programación", McGraw-Hill, 1992.

Brassard, G y Bratley, P: "Algorithmique", Masson, 1987.

Braunstein, Silvia y Gioia, Alicia: "Introducción a la programación y a las estructuras de datos, Eudeba, 1986.

Clavel, Biondi:" Algoritmica y lenguajes", Masson, 1985.

Clavel, Biondi:" Estructuras de datos", Masson, 1985.

Coleman,D: "Organización de datos y programación esturcturada, Gili, 1986.

Courtin, y Kowarski, I: "Introducción a la programación y a las estructuras de datos",
Dunod, 1987.

Dijkstra, E: " A discipline of programming", Prentice Hall, 1976.

Guihur, R: "Procedimientos de clasificación", Masson, 1987.

Harel, David: "Algorithmics", Addison Wesley, 1987.

Joyanes, Luis: "Fundamentos de programación", McGraw-Hill, 1996.

Lewis,T y Smith, M: "Estructuras de datos", Paraninfo, 1985.

Lipschutz, Seymour:" Estructura de datos", McGraw-Hill, 1986.

Rodriguez Almeida, Miguel Angel: "Metodología de la programación", McGraw-Hill, 1993.

Sabatini, Domenico: "Introduzione alla programazione estructurada", Bufeti Editore, 1985.

Wirth, Niklaus: "Introduccion a la programación sistemática", El Ateneo, 1984.

Wirth, Niklaus: "Algoritmos y Estructuras de datos", Prentice Hall, 1987.

Se terminó de imprimir en aries estudio gráfico
Agustín de la Vega 1559 • B1683BXC Martín Coronado • 4973-4359
en Abril del 2000